



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

무선 센서 네트워크에서 노드 연결
밀도 완화를 통한 에너지 효율적인
위치 추정 알고리즘

Energy Efficient Localization by Relaxing
Node-Connection Density in Wireless Sensor
Networks

2016년 2월

서울대학교 대학원

전기 정보 공학부

김 창 회

초 록

로봇이나 센서들이 소형화되고, 센싱 기술과 통신 기술이 발전하면서 low-cost, low-power의 특징을 가진 센서들이 많이 개발되었다. 최근에 이러한 센서 노드들의 다양한 용도에 대해 연구가 활발해지고 있다. 병원에서 환자 나 의사들의 이동을 보거나, 숲에서 일어난 화재의 위치를 파악하거나 군용 로봇의 위치를 파악하는 등 센서 노드에 있어서 어떤 목적으로 쓰이더라도 위치를 추정하는 것은 반드시 포함되어야 하는 기술이 되었다. 본 논문에서는 무선 센서 네트워크 환경에서 센서 노드를 추정함에 있어서 에너지 효율을 높이기 위한 두 가지 알고리즘을 제시한다.

다차원 척도법은 데이터를 종류에 따라 수치화하였을 때, 비슷한 유형의 데이터들을 상대적인 거리로 표현함으로써 데이터들의 성격을 시각화하는 기법이다. 여기서 데이터를 수치화하는 방법으로 노드 사이의 거리를 대입하면 노드 간의 상대적인 거리를 통해 추정 위치를 시각화할 수 있다.

노드의 밀도가 높을 때, 다차원 척도법을 그대로 사용할 경우, 연산량 증가로 인해 측위가 느려지고, CPU 사용도 많아지게 된다. 본 논문에서는 분산 측위 환경에서 이웃 노드(커버리지 안에 1 홉으로 통신이 가능한 노드)의 개수에 따라 송신 전력을 조절하는 알고리즘을 제안한다. 중심 노드만을 고려하여 이웃 노드의 개수에 따라 송신 전력을 이산적으로 조절하는 방법과 가장 바깥쪽 이웃 노드의 Connectivity를 고려하여 전력을 조절하는 방법을 제시한다. 전체 노드 개수에 따른 정확도와 측위에 사용되는 에너지를 시뮬레이션을 통해 비교 및 검증한다.

주요어 : 위치 추정, 다차원척도법, 멀티홉 네트워크, 무선 센서 네트워크

학 번 : 2014-21657

목 차

초 록	i
목 차	ii
그림 목차	iv
표 목차	v
제 1 장 서 론	1
제 2 장 배경이론 및 문제의 정의	4
제 1 절 Received Signal Strength Indicator (RSSI)	4
제 2 절 멀티홉 통신 (Multihop Communication)	5
제 3 절 중앙처리 측위과 분산처리 측위	6
제 3 장 위치 추정 알고리즘	9
제 1 절 삼변 측량법과 최소 제곱법	9
제 2 절 Classical Multidimensional Scaling (CMDS)	12
제 3 절 Distributed-Weighted Multidimensional Scaling (DW-MDS)	15

제 4 장 Energy-Efficient Multidimensional Scaling	19
제 1 절 Discretely Power-Controlled Multidimensional Scaling (DPC-MDS)	19
제 2 절 Edge node Removed Multidimensional Scaling (ER-MDS)	22
 제 5 장 시뮬레이션 결과.....	25
제 1 절 거리 오차를 통한 알고리즘 성능 분석	27
제 2 절 DW-MDS와 제안한 알고리즘의 에너지 소모 비교	31
제 3 절 기준 이웃 노드 개수에 따른 알고리즘 성능 변화	35
제 4 절 제안한 알고리즘과 DW-MDS의 복잡도 비교	39
 제 6 장 결론.....	41
 참고문헌.....	43
 Appendix.....	45
 Abstract	48

표 목차

[표 3-1] DW-MDS에 표기되는 기호의 의미	15
[표 4-1] DPC-MDS에서의 pseudo-code.....	20
[표 5-1] 오차 Variance와 노드 개수에 따른 RMSE 비교.....	30
[표 5-2] 송신 모드에서의 소모 전력 및 에너지.....	32
[표 5-3] 수신 모드에서의 소모 전력 및 에너지.....	33

그림 목차

[그림 1.1] 무선센서네트워크의 구성	2
[그림 2.1] Micaz Radio Module.....	5
[그림 2.2] 중앙처리 측위 시스템	7
[그림 2.3] 분산처리 측위 시스템	8
[그림 3.1] 삼변측량법 개요도	9
[그림 3.2] DW-MDS에서 iteration에 따른 노드의 추정 위치 변화.....	18
[그림 4.1] DPC-MDS의 알고리즘에서의 state 관계	20
[그림 4.2] DPC-MDS를 적용한 중심 노드의 커버리지 변화... 21	
[그림 4.3] ER-MDS를 적용한 중심 노드의 커버리지 변화.....	24
[그림 5.1] 50m X 50m 환경에서 앵커 노드와 타겟 노드	25
[그림 5.2] 15.4m X 22.6m 실내에서의 Pathloss Exponent....	26
[그림 5.3] iteration 횟수에 따른 ER-MDS(왼쪽)와 DPC-MDS(오른쪽)의 RMSE.....	28
[그림 5.4] 알고리즘 별 노드 개수에 따른 RMSE	29
[그림 5.5] 알고리즘 결과 추정된 노드의 위치	30
[그림 5.6] 각 알고리즘에서 소모되는 에너지	33
[그림 5.7] 노드 1개에 대한 평균 송신 에너지	34
[그림 5.8] 이웃 노드 개수 제한에 따른 DPC-MDS와 ER-MDS 의 RMSE 성능 비교.....	36
[그림 5.9] 이웃 노드 개수 제한에 따른 알고리즘의 에너지 소모량 비교	37
[그림 5.10] 이웃 노드 개수 제한이 6, 12일 때 노드 1개의 평균 송신 에너지	38
[그림 5.11] NL=8일 때, 이웃 노드 개수의 변화.....	39
[그림 5.12] NL=8,12일 때, DPC-MDS와 ER-MDS에서의 이웃 노드 개수	40

제 1 장 서 론

로봇이나 센서들이 소형화되고, 센싱 기술과 통신 기술이 발전함에 따라 저비용, 저전력의 특징을 가진 센서들이 개발되었다. 이를 통해 많은 수의 센서를 통한 무선센서네트워크(Wireless Sensor Networks, WSN)를 구축하는 것이 가능해졌다[1]. 그림 1.1은 무선센서네트워크의 구성을 표현한 그림이다. 그림처럼 특정한 사이즈 안에서 통신이 가능한 센서들이 연결을 이루고 있는 네트워크를 의미한다. 최근에 이러한 센서 노드들의 다양한 용도에 대해 연구가 활발해지고 있다. 병원에서 환자나 의사들의 이동을 보거나, 숲에서 일어난 화재의 위치를 파악하거나 군용 로봇의 위치를 파악하는 등 센서 노드에 있어서 어떤 목적으로 쓰이더라도 위치를 추정하는 것은 반드시 포함되어야 하는 기술이 되었다[2].

무선 센서 네트워크의 다양한 활용 용도에도 불구하고 한계로 지적되는 점은 배터리 용량의 제한때문이다. 이에 따라 무선 센서 네트워크에서 에너지 소비를 모델링하고, 적은 에너지 소비를 이용한 네트워크를 구성하는 등의 연구도 진행되었다[3-4]. 센서의 배터리 수명이 곧 무선 센서 네트워크를 구성하고 사용하는 시간과 직결되기 때문에 배터리 소모를 줄이려는 연구가 활발하다. 또한 센서 노드의 많은 에너지가 주변 노드와의 통신에 의해 소모되므로 송신 전력 크기의 최적화는 에너지 소모를 줄이는 데 큰 도움이 될 것이다.

무선 센서 네트워크에서 주변 정보만 이용하는 지, 네트워크 안에 있는 모든 정보를 이용하는 지에 따라 2가지로 측위 방법을 구분할 수 있다. 중앙처리 측위(Centralized Localization)와 분산처리 측위(Distributed Localization)로 구분하는데, 중앙처리 측위란 측위를 하기 위해 필요한 모든 정보를 중앙 시스템에 보내고 이 시스템에서 계산 과정을 거쳐서 위치를 추정하는 방법을 뜻한다[5]. 분산처리 측위는 별도의 연산 장치 없이 자신의 주변 노드의 정보를 이용하여 스스로 연산 과정을 거쳐 자신의 위치를 추정하는 측위 방법을 의미한다. 노드의 수가 적은 무선 센서 네트워크에서는 두 측위 방법 모두 충분히 적용될 수 있지만 노드의 수가 늘어날수록 여러 홉을 거친 모든 노드의 정보를 저장해야하는 over load가 문제가 되기 때문에 중앙처리 측위를 적용하기 힘들어진다. 따라서 많은 수의 노드로 구성되는 네트워크의 경우 분산처리

식 측위 방법이 사용된다.

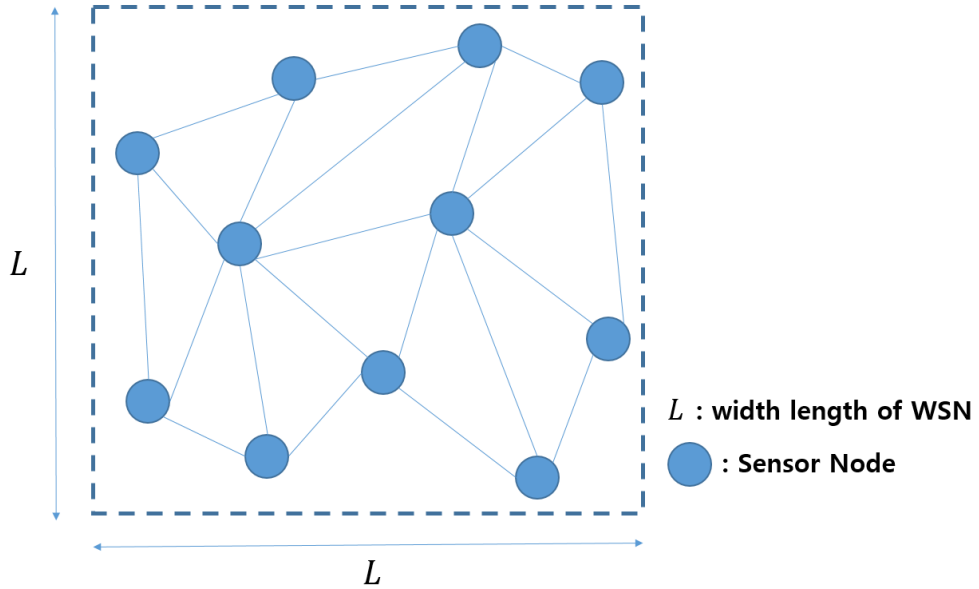


그림 1.1 무선센서네트워크의 구성

본 논문에서는 분산처리식 측위를 이용하는 다차원 척도법 (Multidimensional Scaling, MDS)을 소개한다. 다차원 척도법이란 여러 데이터에 항목 별로 크기를 부여하여 각각의 데이터 간에 유사도를 시각화하는 분석 방법이다[6]. 기존의 MDS는 모든 노드 간의 거리 정보를 이용하여 중앙처리식 측위를 적용하거나 주변 노드의 상황과는 관계없이 거리 정보를 이용함으로써 많은 계산량을 통한 측위를 이용하였다[7]. 그러나 다수의 노드를 사용하는 무선 센서 네트워크에서는 분산처리식 측위가 필요했다[8]. 한편으로 무선 센서 네트워크에서 다수의 노드가 사용된다는 점에서 연결 기반의 측위 알고리즘이 적용한 연구도 진행되었다[9]. 그러나 이러한 알고리즘들은 노드의 밀도에 따른 무선 센서 네트워크의 특징을 활용하지 못하였다. 본 논문에서는 주변 노드의 개수 (정보량)에 따라 센서의 송신 전력을 조절함으로써 에너지 소비를 줄이는 측위 알고리즘을 제안한다. 송신하는 노드 스스로 주변 노드의 개수를 파악하여 이산적으로 전력을 조절하는 알고리즘과 가장 자리에 있는 노드의 연결 정보를 파악하여 전력을 조절하는 알고리즘 2가지를 제안한다. 본 논문에서 제안하는 알고리즘을 기존의 측위 알고리즘과 거리오차, 소모한 에너지에 대해서 비교 및 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 배경이론과 본 논문에서 정의하고 있는 문제에 대해서 살펴본다. RSSI를 통해 거리를 측정하는

방법과 위치 추정을 분류하는 방법 중에서 중앙처리식 측위와 분산처리식 측위에 대해서 알아본다. 3장에서는 본 논문의 알고리즘과 비교할 위치 추정 방법들에 대해서 소개한다. 3개 이상의 앵커노드를 이용해서 교점을 찾음으로써 타겟 노드의 위치를 추정하는 삼변측량법, MDS 중에서도 모든 노드와의 거리를 추정하여 중앙처리식 측위로 처리하는 Classical MDS(CMDS), 노드 주변의 이웃 노드와의 거리 정보만으로 MDS를 이용하는 DW-MDS에 대해 설명한다. 4장에서는 본 논문에서 제안하는 알고리즘인 Discretely Power-Controlled MDS(DPC-MDS)와 Edge node Remove MDS(ER-MDS)에 대해서 설명한다. 5장에서는 3장에서의 측위 알고리즘과 제안한 알고리즘을 시뮬레이션을 통해서 비교 및 분석한다. 마지막으로 6장에서는 결론을 맺는다.

제 2 장 배경이론

제 1 절 Received Signal Strength Indicator (RSSI)

RSSI 방식은 수신단에서 수신 신호를 감지하고, 그 세기를 이용하여 송신단과의 거리를 추정하는 방식이다. dBm으로 표기되는 신호의 세기를 감지하고, Pathloss Model에 대입하여 거리를 계산한다. 신호를 보낸 시간과 받은 시간의 차이를 이용하여 거리를 추정하는 Time-of-Arrival(ToA)에 비해서 정확도가 떨어지지만, 하드웨어 구현이 쉽다는 장점이 있다. ToA는 수신단과 송신단이 서로 시간에 대해 동기화 되어 있어야만 신호가 수신되기까지의 시간을 계산할 수 있다.

i 노드에서 신호를 송신해서 j 노드에서 받은 신호의 크기 PL_{ij} 를 아래의 Pathloss Model로 계산하여 추정 거리(d_{ij})를 얻어낼 수 있다.

$$PL_{ij} = PL_0 - 10 \cdot n \cdot \log_{10} \left(\frac{d_{ij}}{d_0} \right) + X_s \quad (2.1)$$

여기서 PL_0 은 단위거리 d_0 에서 측정한 신호의 세기를 의미한다. n 은 측위 환경에서 실험을 통해 측정된 path loss exponent이다. X_s 는 small scale fading에 의한 에러이다. Pathloss exponent 값은 측정 환경에 따라서 달라진다. 크게 실내와 실외 환경, LOS와 NLOS 환경으로 구분해서 측정할 수 있다. Free space에서 2를 가지며, 보통 2~4 사이의 값을 가진다. Pathloss Model을 통해 얻어낸 추정 거리는 다음의 식으로 귀결된다.

$$d_{ij} = d_0 \cdot 10^{\frac{PL_0 - PL_{ij}}{10n}} \quad (2.2)$$

본 논문에서는 추정 거리 계산에 필요한 PL_0, n 을 ZigBee 시스템을 통한 실험으로 얻어내었다. ZigBee 시스템은 매우 간단한 하드웨어로 구성되어 있기 때문에 ToA의 사용에는 적합하지 않다. 간단한 것에 비해서 정확도가 떨어질 수 있지만 이것은 여러 노드의 사용을 통해 정보량을 늘려 정확도를 개선하게 된다.

실험에 사용된 모듈은 Kmote Radio Module를 사용하였다. Kmote는 센서 네트워크에서 이용하는 초저전력 무선 노드이다. 가장 기본적인 모듈로서 다양한 센싱 정보를 얻을 수 있다. Kmote의 운영체제는 Tiny OS 오픈 소스 소프트웨어를 통해 동작하고 있다. 250kbps의 전송속도를 가지며, 2.4GHz IEEE 802.15.4 무선 통신 칩셋을 사용한다. 칩셋으로는 CC2420 Radio 칩셋을 사용하고, IEEE 802.15.4을 호환하는 MPR2400CB 모델을 사용하였다. USB 인터페이스를 통해서 RSS 데이터를 수집한 후에 pathloss model에 적용하여 분석하였다.



그림 2.1 Kmote Radio Module.

이 모듈은 최소 -90dBm 까지의 수신 신호 세기까지 감지할 수 있으며, 이는 pathloss exponent가 3인 환경에서 대략 14.68m 정도의 커버리지를 가지는 Sensitivity이다. 따라서 14.68m 이상의 실내 크기에서는 송신단과 수신단이 각자의 위치를 파악하지 못할 수 있다. 이에 따라 더 넓은 크기의 측위 환경에서는 멀티홉 통신을 필수적으로 이용하게 된다.

제 2 절 멀티홉 통신(Multihop Communication)

하나의 송신단에서 하나의 수신단으로 신호를 전달할 때, 이 한번의

과정을 홉(hop)이라고 한다. 송신단에서 수신단으로 정보를 보낼 때, 여러 홉을 거쳐서 통신하는 것을 멀티홉 통신이라고 한다. 이 경우, 송신 노드와 수신 노드 사이에 신호를 전달하는 중간 노드가 존재하게 된다.

멀티홉 통신에서는 송신단과 수신단 사이의 거리를 추정할 때, 송수신 단 사이에 직접 경로의 loss를 알 수 없기 때문에 pathloss model을 사용하지 못한다. 따라서 hop의 개수를 이용하여 대략적인 추정 거리를 계산한다. 송신단 i 노드와 수신단 j 노드 사이의 홉의 개수를 h_{ij} 라고 정의하면 i 노드와 j 노드 사이의 거리 d_{ij} 는 다음과 같이 정의할 수 있다.

$$d_{ij} = h_{ij} * hopdist_{ij} \quad (2.3)$$

h_{ij} 는 i 노드와 j 노드 사이의 홉 개수 중에서 가장 작은 홉의 수를 의미하며, 이것을 계산하는 방법으로 Dijkstra 알고리즘을 사용하였다. Dijkstra 알고리즘은 source에서 destination까지 신호를 전송할 때마다 노드에서 최소 홉을 계산하면서 나아가는 알고리즘을 의미한다. 최종적으로 destination에 도착했을 때 최소 홉을 얻을 수 있다[10]. $hopdist_{ij}$ 는 i 노드와 j 노드 사이의 1홉의 거리를 의미한다. $hopdist_{ij}$ 는 일반적인 경우 노드의 신호 커버리지 거리로 계산한다.

제 3 절 중앙처리 측위과 분산처리 측위

멀티홉 통신에서의 측위 시스템에선 중앙처리 측위(Centralized Localization)과 분산처리 측위(Distributed Localization)이 존재한다. 중앙처리 측위란 측위를 하기 위해 필요한 모든 정보를 중앙 시스템에 보내고 이 시스템에서 계산 과정을 거쳐서 위치를 추정하는 방법을 뜻한다. 아래의 그림 2.2를 보면 빨간점은 실제 위치를 알고 있는 앵커 노드(Anchor node)이고, 파란점은 추정해야 하는 노드(Target node)를 의미한다. RSSI를 이용해 얻은 거리 정보, 멀티홉을 이용해 얻은 거리 정보 등을 컴퓨터에 모두 전달해서 컴퓨터에서 이 정보들을 수식에 대입하여 모든 노드의 위치 정보를 파악한다. 이 측위는 연산량이 많을 때

수행할 수 있다는 장점이 있지만 컴퓨터와 정보를 주고 받을 노드가 반드시 필요하다는 단점이 있다. 중앙처리 위치 추정을 사용하는 알고리즘에는 DV-Hop 알고리즘과 Classical Multidimensional Scaling(CMDS) 방법, 멀티홉 환경에서의 삼변측량법 등이 있다.

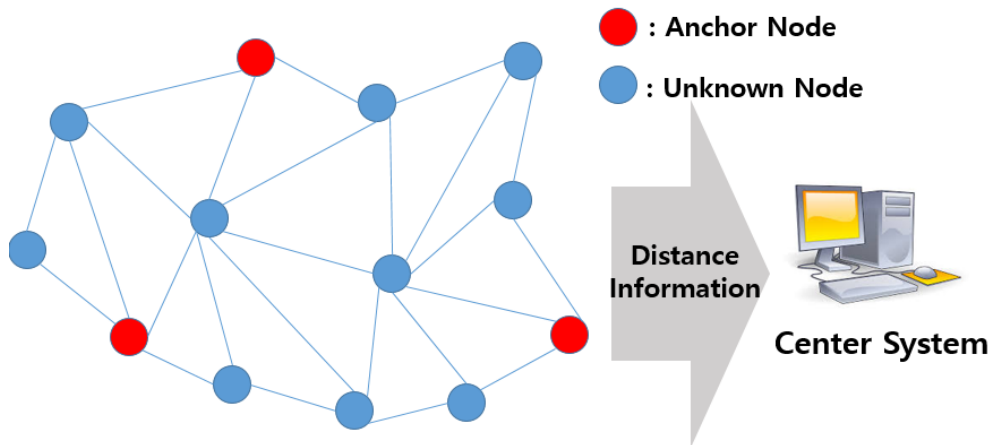


그림 2.2 중앙처리 측위 시스템

분산처리 측위는 별도의 연산 장치 없이 자신의 주변 노드의 정보를 이용하여 스스로 연산 과정을 거쳐 자신의 위치를 추정하는 측위 방법을 의미한다. 자신의 커버리지 범위 안에 있는 노드를 이웃 노드 (Neighborhood node)라고 한다. 이웃 노드들과의 거리 정보를 이용하여 위치를 추정한다. 아래의 그림 2.3은 분산처리 측위 시스템을 간략하게 나타낸 그림이다. 회색 점선은 노드 A의 커버리지 범위를 나타내며, 빨간점은 위치를 알고 있는 앵커 노드, 파란점은 위치를 추정해야 하는 노드를 의미한다. 커버리지 안에 있는 모든 노드는 이웃노드가 되며, A와 이웃 노드들 간의 거리 정보를 A가 추정해서 A 노드가 스스로 연산 과정을 거쳐 자신의 위치가 어디에 있는 지 추정하게 된다. 중앙 처리 장치에 별도로 정보를 전달하는 과정이 없어서 빠르게 추정할 수 있다는 장점이 있다. 하지만 연산량이 많아질 경우 과부하에 걸릴 수 있다는 단점이 있다. 분산처리 측위를 사용하는 알고리즘에는 Distributed Weighted Multidimensional Scaling(Dw-MDS)와 1개의 홉 안에 모든 노드가 존재하는 싱글홉 통신 환경에서의 삼변측량법 등이 있다.

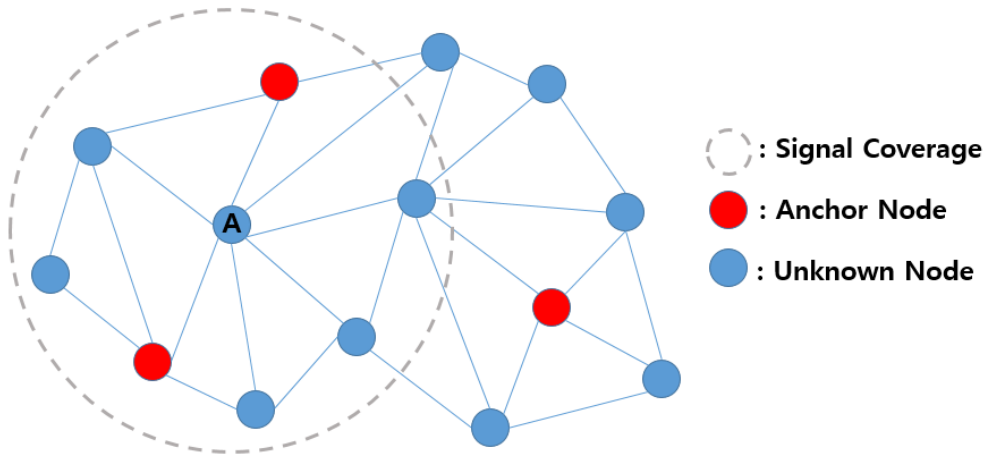


그림 2.3 분산처리 측위 시스템

제 3 장 위치 추정 알고리즘

제 1 절 삼변 측량법과 최소 제곱법

삼변측량은 3개의 Anchor node를 이용하여 Target node의 위치를 추정하는 방법 중 하나이다. 여기서 Anchor node는 위치를 알고 있는 node를 뜻하며, Target node는 측위하고자 하는 node를 의미한다. RSS 오차가 없다고 가정할 때, Target node와 3개의 Anchor node 사이의 거리를 이용해 3개의 원을 만들면 반드시 한 점에서 3개의 원이 만나게 된다. 아래 그림에서 확인할 수 있듯이 Real Range를 이용한 원은 한 점에서 만난다. 이 접점을 Target node의 위치로 추정하게 된다. 그러나 RSS 오차를 포함한 Estimated Range를 형성하는 경우, Target node의 위치를 정확하게 찾을 수 없다.

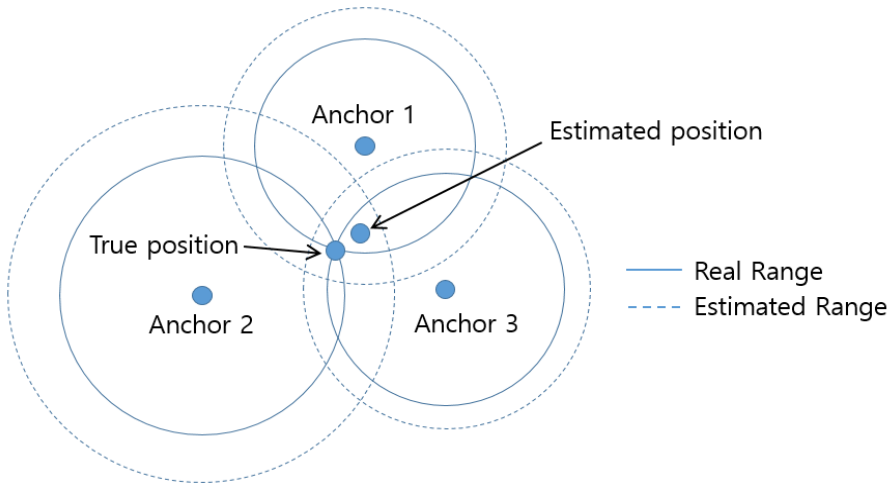


그림 3.1 삼변측량법 개요도

삼변 측량법을 수학적으로 살펴보면 다음과 같다. Anchor 1,2,3의 Estimated range 크기를 각각 $\hat{r}_1, \hat{r}_2, \hat{r}_3$ 라 하면, Anchor node와 Target node 간의 거리를 이용한 원의 방정식은 아래의 수식처럼 표현할 수 있다.

$$\begin{aligned}
(x_1 - x)^2 + (y_1 - y)^2 &= \hat{r}_1^2 \\
(x_2 - x)^2 + (y_2 - y)^2 &= \hat{r}_2^2 \\
(x_3 - x)^2 + (y_3 - y)^2 &= \hat{r}_3^2
\end{aligned} \tag{3.1}$$

(x, y) 는 Target node의 좌표, $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ 는 각각 Anchor node 1,2,3의 좌표를 의미한다. 식 (3.1)에서 첫 번째 식에서 두 번째 식과 세 번째 식을 빼면 다음과 같은 직선의 방정식을 얻을 수 있다.

$$\begin{aligned}
(x_2 - x_1)x + (y_2 - y_1)y &= \frac{1}{2}[(\hat{r}_1^2 - \hat{r}_2^2) + (x_2^2 - x_1^2) + (y_2^2 - y_1^2)] \\
(x_3 - x_1)x + (y_3 - y_1)y &= \frac{1}{2}[(\hat{r}_1^2 - \hat{r}_3^2) + (x_3^2 - x_1^2) + (y_3^2 - y_1^2)]
\end{aligned} \tag{3.2}$$

위의 식을 행렬로 나타내면 아래와 같은 식으로 표현된다.

$$\begin{bmatrix} (x_2 - x_1) & (y_2 - y_1) \\ (x_3 - x_1) & (y_3 - y_1) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (\hat{r}_1^2 - \hat{r}_2^2) + (x_2^2 - x_1^2) + (y_2^2 - y_1^2) \\ (\hat{r}_1^2 - \hat{r}_3^2) + (x_3^2 - x_1^2) + (y_3^2 - y_1^2) \end{bmatrix} \tag{3.3}$$

좌항과 우항의 식을 정리하기 위해 $H = \begin{bmatrix} (x_2 - x_1) & (y_2 - y_1) \\ (x_3 - x_1) & (y_3 - y_1) \end{bmatrix}$, $p = \begin{bmatrix} x \\ y \end{bmatrix}$, $b = \frac{1}{2} \begin{bmatrix} (\hat{r}_1^2 - \hat{r}_2^2) + (x_2^2 - x_1^2) + (y_2^2 - y_1^2) \\ (\hat{r}_1^2 - \hat{r}_3^2) + (x_3^2 - x_1^2) + (y_3^2 - y_1^2) \end{bmatrix}$ 라 하면 아래와 같은 행렬의 식으로 표현할 수 있고, 역행렬을 통해서 position vector p 를 구할 수 있게 된다.

$$\begin{aligned}
Hp &= b \\
p &= H^{-1}b
\end{aligned} \tag{3.4}$$

그런데 만일 Anchor node가 4개 이상이 되어서 식 (3.3)와 같은 연립방정식이 3개 이상이 되어버리면 간단한 행렬 연산으로 구하지 못한다. 거리 오차까지 존재하는 경우에는 모든 연립방정식을 만족하는 해를 구하지 못하게 된다. 이런 경우에 오차를 가장 최소화하면서 해를

구할 수 있는 방법이 최소 제곱법(Least Square)이다. 마찬가지로 Target node의 위치를 (x, y) , 총 N 개의 Anchor node 중에서 i 번째 Anchor node의 위치를 (x_i, y_i) , RSS를 통해 얻은 Target node와 Anchor node 사이의 추정 거리를 \hat{r}_i 라 하면 아래와 같은 원의 방정식들로 표현할 수 있다.

$$\begin{aligned}(x - x_1)^2 + (y - y_1)^2 &= \hat{r}_1^2 \\(x - x_2)^2 + (y - y_2)^2 &= \hat{r}_2^2 \\&\vdots \\(x - x_N)^2 + (y - y_N)^2 &= \hat{r}_N^2\end{aligned}\tag{3.5}$$

위 식에서 N 번째 원의 방정식을 다른 원의 방정식에서 빼서 $(N - 1)$ 개의 식으로 표현하면 다음과 같은 직선의 방정식으로 나타난다.

$$\begin{aligned}(x_1 - x_N)x + (y_1 - y_N)y &= \frac{1}{2}[(\hat{r}_1^2 - \hat{r}_N^2) + (x_N^2 - x_1^2) + (y_N^2 - y_1^2)] \\(x_2 - x_N)x + (y_2 - y_N)y &= \frac{1}{2}[(\hat{r}_2^2 - \hat{r}_N^2) + (x_N^2 - x_2^2) + (y_N^2 - y_2^2)] \\&\vdots \\(x_{N-1} - x_N)x + (y_{N-1} - y_N)y &= \frac{1}{2}[(\hat{r}_{N-1}^2 - \hat{r}_N^2) + (x_N^2 - x_{N-1}^2) + (y_N^2 - y_{N-1}^2)]\end{aligned}\tag{3.6}$$

위의 식은 아래와 같이 선형 행렬의 식으로 표현 가능하다.

$$Hp = b\tag{3.7}$$

추정된 Target node의 위치를 \hat{p} 라고 하면 추정된 위치와 측정값 사이의 오차 ε 는 다음과 같이 표현할 수 있다.

$$\varepsilon = b - H\hat{p}\tag{3.8}$$

최소제곱법은 ε 을 가장 최소화 시키는 \hat{p} 을 찾는 방법을 의미한다. ε 을 제곱한 것을 비용함수 J 라 하면 J 를 \hat{p} 에 대해 미분하여 0이 되는 지점이 이차 함수 J 를 가장 최소화 시키는 지점이 된다. 비용함수 J 는 다음과 같다.

$$\begin{aligned} J &= \varepsilon^T \varepsilon = (b - H\hat{p})^T (b - H\hat{p}) \\ &= b^T b - 2(H^T b)^T \hat{p} + \hat{p}^T H^T H \hat{p} \end{aligned} \quad (3.9)$$

식 (3.8)은 \hat{p} 에 대한 2차 함수이고, 이를 \hat{p} 에 대해 편미분을 하면 아래와 같은 식으로 표현된다.

$$\frac{\partial J}{\partial \hat{p}} = 2H^T H \hat{p} - 2H^T b \quad (3.10)$$

식 (3.10)가 0이 되는 \hat{p} 을 구하면 아래의 식 (3.11)처럼 표현할 수 있고, \hat{p} vector의 (x, y) 값을 통해서 Target node의 위치를 찾을 수 있다.

$$\hat{p} = (H^T H)^{\dagger} H^T b \quad (3.11)$$

H 는 정방행렬이 아니므로, 역행렬이 존재하진 않지만 pseudo-inverse를 통해서 역행렬을 취할 수 있다. 최종적으로 pseudo-inverse를 통해 Target node의 위치를 추정하게 된다.

제 2 절 Classical Multidimensional Scaling (CMDS)

다차원 척도법 (Multidimensional Scaling, MDS)이란 여러 데이터에 항목 별로 크기를 부여하여 각각의 데이터 간에 유사도를 시각화하는 분석 방법이다. 데이터를 시각화할 수 있는 장점 때문에 경제학, 사회학, 행동 과학 등의 분야에 많이 응용된다. 측위 시스템에서는 데이터의 종류가 노드 사이의 거리가 되며, 노드 간의 유사도를 시각화하는 과정이 위치를 추정하는 과정이 된다. 다시 말해서, 다차원 척도법은 다른 노드와의 거리 정보를 통해서 노드 자신의 상대적인 위치를 알아내는 방법이다. 최근에는 beacon을 이용한 센서 네트워크에도 이 기술을 적용한 연구도 진행되고 있다[11]. 다양한 MDS 방법 중에서 Classical Multidimensional Scaling (CMDS)는 노드 간 거리 정보로 이루어진 행렬을 특이값 분해(Singular Value Decomposition, SVD)를 하여 Closed formed solution을 얻게 되고, 고유 벡터(Eigen vector)를

통해 2차원 혹은 3차원 평면에 위치를 나타낼 수 있다.

i 번 노드와 j 번 노드 사이의 측정 거리를 $\{\delta_{ij}\}_{i,j}^N$ 라 정의하면 $\{\delta_{ij}\}_{i,j}^N$ 는 i 번 노드와 j 번 노드의 좌표 성분을 이용해 아래의 식처럼 표현할 수 있다.

$$\{\delta_{ij}\}_{i,j}^N = \|\hat{p}_i - \hat{p}_j\| = \sqrt{(\hat{p}_i - \hat{p}_j)^T (\hat{p}_i - \hat{p}_j)} \quad (3.12)$$

N 은 전체 노드의 개수를 의미한다. 식 (3.12)에서 \hat{p}_i 은 i 노드의 2차원 위치 벡터를 의미하며, x 좌표와 y 좌표를 통해 $\hat{p}_i = [\hat{x}_i \ \hat{y}_i]^T$ 로 표현할 수 있다. 원점에서 노드까지의 거리의 제곱을 벡터로 표현하면 다음과 같이 나타낼 수 있다.

$$\psi = [\hat{p}_1^T \hat{p}_1, \hat{p}_2^T \hat{p}_2, \dots, \hat{p}_N^T \hat{p}_N]^T \quad (3.13)$$

식 (3.13)에 의해 각 노드 간 거리의 제곱 성분으로 이루어진 행렬 D 는 아래처럼 표현할 수 있다.

$$D = [d_{ij}^2]_{i,j=1}^N = \psi e^T - 2P^T P + e \psi^T \quad (3.14)$$

식 (3.14)에서 e 는 모든 성분이 1이고 크기가 N 인 벡터이다. P 는 모든 노드의 위치 벡터를 나열한 행렬로 $P = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_N]$ 처럼 나타낼 수 있다. 행렬 성분의 평균 성분을 빼주는 Centering operator를 H 라 하면 $H = I - ee^T/N$ 이고,

$$B = -\frac{1}{2}HDH = HP^T PH \quad (3.15)$$

B 는 $P^T P$ 에 H 를 곱함으로써 열벡터의 평균을 0으로 만들어준다. B

행렬은 D 행렬과 더붙어서 모든 노드와의 거리 정보를 담고 있어야 한다. 그리고 B 행렬을 특이값 분해(Singular Value Decomposition, SVD)를 하여 X의 좌표를 복원한다. B를 SVD하면 2차원 평면에서는 최소 랭크(rank)의 크기가 2까지 3차원 좌표의 데이터라면 랭크의 크기가 최소 3인 고유값의 행렬을 얻을 수 있다. 이를 수식적으로 표현하면

$$B = U \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{dim}) U^T = P_{rel}^T P_{rel} \quad (3.16)$$

식 (3.16)에서 U 는 고유벡터를 의미하며, $\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{dim})$ 는 $\lambda_1, \lambda_2, \dots, \lambda_{dim}$ 을 대각 성분으로 가지는 dim-by-dim 크기의 행렬이다. dim은 노드가 존재하는 차원을 의미한다. 본 논문에서는 2차원 평면에 노드가 존재하였다고 가정한다. (dim=2) P_{rel} 은 상대적인 노드의 위치 벡터들로 이루어진 행렬이다. 식 (3.16)에 의해 P_{rel} 은 다음과 같이 구할 수 있다.

$$P_{rel} = \text{diag}((\lambda_1^{1/2}, \lambda_2^{1/2}, \dots, \lambda_{dim}^{1/2}) U^T \quad (3.17)$$

이것을 실제 위치에 맵핑(mapping)하기 위해서 Procrustes 분석에 대입한다. Procrustes 분석에서는 유사한 두 데이터에 대해서 회전, 비례, 평행이동을 수행하여 두 데이터 간의 차이를 최소화하는 위치로 맵핑해준다. 여기서 P_{rel} 안에 있는 Anchor node의 위치 정보와 실제 Anchor node의 위치 정보를 Procrustes 분석에 대입한다. Procrustes 분석에 의해 구한 평행이동, 회전, 비례에 대한 벡터를 각각 k, r, c 라 하면 CMDS에 의해 추정된 위치는 다음과 같다.

$$\tilde{P} = c \cdot r^T P_{rel} + k \quad (3.18)$$

이와 같이 모든 노드 간의 거리 행렬과 Procrustes 분석에 의해 모든 노드의 위치 벡터를 얻는 방법을 Classical Multidimensional Scaling

(CMDS)라고 한다. 이 방법은 어떤 한 시스템에서 모든 노드 사이의 거리 정보를 얻을 수 있어야 노드들의 위치를 추정할 수 있기 때문에 중앙처리 측위 시스템으로 볼 수 있다.

제 3 절 Distributed-Weighted Multidimensional Scaling (DW-MDS)

DW-MDS는 CMDS와는 달리 분산처리 측위 방식을 이용한 다차원 척도법이다. 분산처리 측위 방식은 모든 노드와의 거리 정보 없이 중심 노드의 주변에 존재하는 노드의 정보만으로 중심 노드 자신의 위치를 스스로 추정하는 방식이다. DW-MDS의 위치 추정 방식은 전체 비용 함수를 줄여 나감으로써 비용함수를 최소로 만드는 위치를 찾는 방식이다.

$$S = 2 \sum_{1 \leq i \leq n} \sum_{i < j \leq n+m} w_{ij} (\delta_{ij} - d_{ij}(P))^2 \quad (3.19)$$

식 (3.19)의 기호들은 아래의 표 (3.1)의 정의를 따른다.

표 3.1 DW-MDS에 표기되는 기호 의미

기호	의미 설명
S	전체 비용 함수
S_i	i 노드를 중심 노드로 했을 때의 비용함수
$N = n + m$	전체 센서 노드의 개수
n	위치를 모르는 센서 노드의 개수 (추정 대상)
m	위치를 알고 있는 앵커 노드의 개수
P	모든 노드의 실제 좌표 벡터를 나열한 행렬
$P^{(k)}$	k 번째 iteration일 때, 모든 노드의 좌표 벡터를 나열한 행렬
p_i	i 노드의 좌표 벡터, $(x_i, y_i)^T$
$p_i^{(k)}$	k 번째 iteration일 때, i 노드의 좌표 벡터 $(x_i^{(k)}, y_i^{(k)})^T$

w_{ij}	i 노드와 j 노드 사이의 가중치
$d_{ij}(P)$	좌표 행렬 P 에 의해 계산한 i 노드와 j 노드 사이의 거리(= $d_{ji}(P)$)
δ_{ij}	i 노드와 j 노드 사이의 측정 거리(= δ_{ji}) (by RSS)

식 (3.19)의 전체 비용함수는 측정 거리와 실제 거리의 차이를 제공한 다음 이것을 가중치를 곱해줘서 i 노드와 j 노드 사이의 비용함수를 모든 노드에 취해줌으로써 계산할 수 있다. 여기서 가중치 w_{ij} 는 다음과 같은 식을 통해 계산한다.

$$w_{ij} = \begin{cases} \exp\left(-\frac{\delta_{ij}^2}{h_{ij}^2}\right) & , \text{ if } \delta_{ij} \text{ is measured} \\ 0 & , \text{ otherwise} \end{cases} \quad (3.20)$$

식 (3.20)에서 h_{ij} 는 $\max(\text{coverage}_i, \text{coverage}_j)$ 로 i 노드 혹은 j 노드의 coverage 중에서 큰 coverage에 해당하는 크기이다. 따라서 $h_{ji} = h_{ij}$ 가 되므로 $w_{ij} = w_{ji}$ 가 된다. 가중치는 통신 가능한 거리 대비 얼마나 가까운 곳에 위치하는 가를 의미하는 변수이다. 측정 범위가 넓고 측정 거리가 가까울수록 정확한 데이터라고 할 수 있기 때문에 가중치의 크기가 커진다. 그리고 i 노드와 j 노드 사이에 측정값이 없으면 정확도가 0이므로 가중치 $w_{ij} = 0$ 이 된다.

모든 노드에 대해서 각각 독립적인 비용함수가 존재한다고 하면, 전체 비용함수는 모든 노드의 비용함수의 합으로 생각할 수 있다.

$$S = \sum_{i=1}^n S_i \quad (3.21)$$

각각의 독립적인 비용함수들의 합이 전체 비용함수 S 가 되기 위해서 S_i 는 아래와 같이 정의할 수 있다.

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} \left(\delta_{ij} - d_{ij}(P) \right)^2 + \sum_{j=n+1}^{n+m} 2w_{ij} \left(\delta_{ij} - d_{ij}(P) \right)^2 \quad (3.22)$$

분산처리 측위를 하기 위해선 부분 비용함수 S_i 가 최소가 되는 $d_{ij}(P)$ 를 찾아야 한다. $\frac{dS_i}{dp_i} = 0$ 인 p_i 를 구함으로써 i 노드의 위치 벡터 p_i 에 대해 S_i 가 최소가 되는 지점을 구하게 된다. 다만, CMDS와는 달리 DW-MDS는 Closed-form의 해를 구할 수 없다. SMACOF (Scaling by MAjorizing a Complicated Function) 알고리즘의 majorizing 함수를 이용하여 반복적으로 해를 구함으로써 최적의 위치벡터 p_i 를 구할 수 있다. Appendix A를 참고하여, k 번째 iteration에서 i 노드의 위치벡터 $p_i^{(k)}$ 를 아래와 같이 상수 a_i 와 이전 iteration의 위치 행렬 $P^{(k-1)}$, 이전 iteration의 $b_i^{(k-1)}$ 벡터의 곱으로 얻을 수 있다.

$$\begin{aligned}
 p_i^{(k)} &= a_i(P^{(k-1)}b_i^{(k-1)}) \\
 a_i^{-1} &= \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} + \sum_{j=n+1}^{n+m} 2w_{ij} \\
 b_i^{(k-1)} &= [b_1^{(k-1)}, b_2^{(k-1)}, \dots, b_i^{(k-1)}, \dots, b_{n+m}^{(k-1)}]^T \\
 b_i^{(k-1)} &= \begin{cases} b_j = w_{ij}[1 - \delta_{ij}/d_{ij}(P^{(k-1)})], & j \leq n, j \neq i \\ b_i = \sum_{j=1}^n \frac{w_{ij}\delta_{ij}}{d_{ij}(P^{(k-1)})} + 2 \sum_{j=n+1}^{n+m} \frac{2w_{ij}\delta_{ij}}{d_{ij}(P^{(k-1)})} \\ b_j = 2w_{ij}[1 - \delta_{ij}/d_{ij}(P^{(k-1)})], & j > n \end{cases}
 \end{aligned} \tag{3.23}$$

위의 과정을 반복적으로 하여 최적의 위치를 찾는 것이 DW-MDS의 목표이다. 식 (3.23)처럼 i 노드의 해를 구한 후에 $(i+1)$ 노드의 해를 구하고, 마지막으로 n 노드의 해까지 구하는 과정을 끝냄으로써 k 번째 iteration을 모두 수행하게 된다. 그림 3.2와 같이 iteration이 증가할수록 비용함수를 최소화 시키는 위치로 추정 위치를 변화시키게 된다. Iteration의 개수 k 를 정하는 것은 전체 비용함수가 줄어드는 크기가 특정값 ϵ 보다 작아질 때, iteration을 중단하게 된다. ϵ 의 값은 사용자가 임의로 정하는 상수이며, 크기가 클수록 적은 iteration으로 노드의 위치를 추정하게 된다.

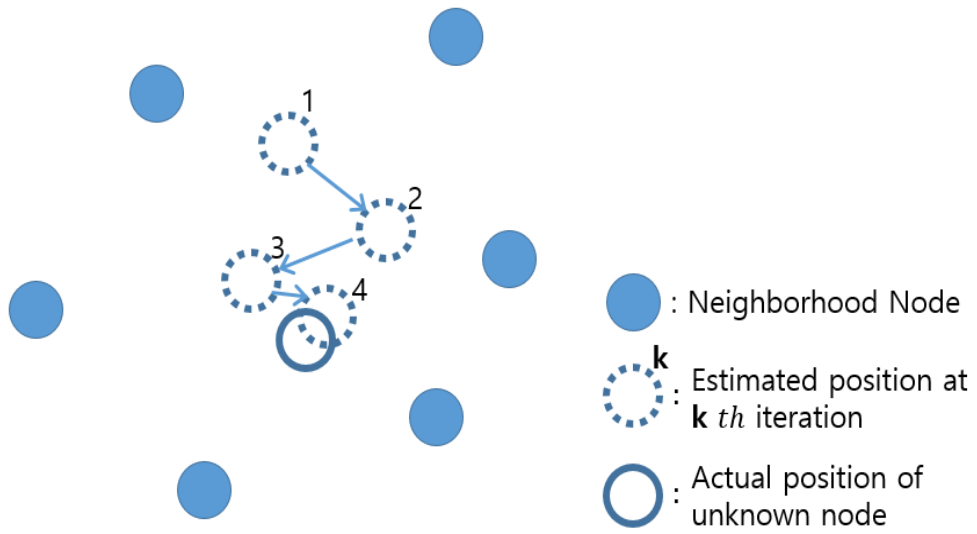


그림 3.2 DW-MDS에서 iteration에 따른 노드의 추정 위치 변화

제 4 장 Energy-Efficient Multidimensional Scaling

제 4 장에서는 본 논문에서 제안하는 알고리즘인 에너지 효율적인 다차원 척도법에 대해 제시한다. 노드의 송신 전력을 이웃 노드의 개수에 따라서 이산적으로 조절하는 Discretely Power-Controlled Multidimensional Scaling(DPC-MDS) 방법과, 가장 바깥 노드의 Connectivity를 고려하여 송신 전력을 연속적으로 조절하는 Edge node Removed Multidimensional Scaling(ER-MDS) 방법을 제안한다.

제 1 절 Discretely Power-Controlled Multidimensional Scaling (DPC-MDS)

DPC-MDS는 이웃 노드의 개수에 따라서 단계적으로 송신 전력을 줄일 수 있는 알고리즘이다. 송신 전력을 줄이는 정도는 연속적으로 조절하는 것이 아니라 사용자가 정한 전력만큼 이산적으로 전력을 줄이게 된다. 모든 노드의 송신 전력의 크기는 3가지 State를 통해 결정된다. 이웃 노드의 개수에 따라 State가 바뀌면서 송신 전력을 조절하게 된다.

$$TP_j = \begin{cases} TP_2 (dbm) & \text{where State1 and } neigh_i \geq NL \\ TP_1 (dbm) & \text{where State0 and } neigh_i \geq NL \\ & \text{State2 and } neigh_i < NL \\ TP_0 (dbm) & \text{where state1 and } neigh_i < NL \\ & \text{when the algorithm starts} \end{cases} \quad (4.1)$$

식 (4.1)에서 TP_0, TP_1, TP_2 는 각각 State 0, State 1, State 2에서의 송신 전력(Transmit Power)을 의미한다. NL (Neighborhood nodes Limit)은 전력을 조절하기 위해 최소한으로 보장되어야 하는 이웃 노드의 개수를 의미한다. State 0, State 1, State 2는 아래의 그림 4.1에서 구체적으로 설명되어 있다. State 0은 초기 상태의 state를 의미하는 단계로써 모든 노드에서 송신 전력을 조절하지 않고 초기에 설정된 송신 전력으로 전송하는 단계를 의미한다. State 1은 State 0에서 i 노드의 이웃 노드의 개수 $neigh_i$ 가 NL 보다 크거나 State 2에서 NL 보다 이웃 노드 개수가 적

을 때 적용되는 단계이다. 이 때의 i 노드는 TP_1 크기의 송신 전력을 이용한다. State 2에서는 State 1에서도 $neigh_i$ 가 NL 보다 크면 적용되는 단계로, TP_2 크기의 송신 전력을 이용한다.

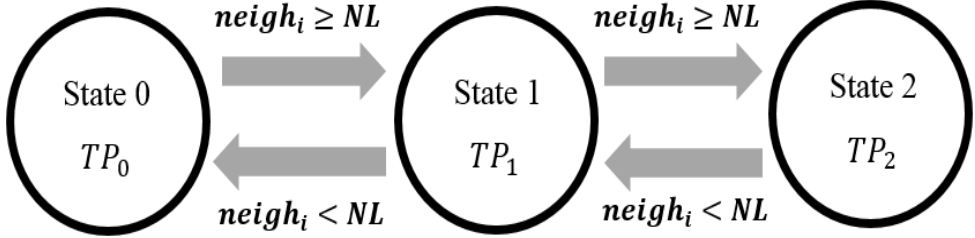


그림 4.1 DPC-MDS 알고리즘에서의 State 관계

State가 바뀌는 과정이 iteration 과정에 삽입되어 한번의 iteration에서 모든 노드에 위의 알고리즘이 실행된다. Iteration 횟수는 localization time에 따라 고정하였다. 본 논문에서는 거리 오차를 3m 이하를 보장하는 iteration 횟수를 위해 총 iteration 횟수는 8회 혹은 12회로 고정하였다. 이를 pseudo-code로 나타내면 아래와 같다.

표 4.1 DPC-MDS의 pseudo-code

```

Initial State  $\rightarrow k = 0, p_i^{(0)}, a_i$  is obtained
For iteration=1 to the number of iteration
   $k \leftarrow k + 1$ 
  for  $i=1$  to  $n$ 
    compute  $b_i^{(k-1)}$ 
     $p_i^{(k)} = a_i(P^{(k-1)}b_i^{(k-1)})$ 
    Case (TP of  $i$  node)
       $TP_0 \rightarrow$  if  $neigh_i > NL$ 
        TP of  $i$  node =  $TP_1$ 
      end
       $TP_1 \rightarrow$  if  $neigh_i > NL$ 
        TP of  $i$  node =  $TP_2$ 
  
```

```

else
    TP of i node =  $TP_0$ 
end
 $TP_2 \rightarrow$  if  $neigh_1 \leq NL$ 
    TP of i node =  $TP_1$ 
end
endcase
weighting factor  $w_{ij}$  updated
end for
end for

```

State가 높아짐에 따라 송신 전력의 크기가 작아지고, 노드의 통신 커버리지는 좁아진다. 아래의 그림 4.2에서 r_0, r_1, r_2 는 각각 TP_0, TP_1, TP_2 에 의한 커버리지의 반지름이다. 중심 노드의 주변에 있는 이웃 노드의 개수가 점점 적어지는 것을 알 수 있다.

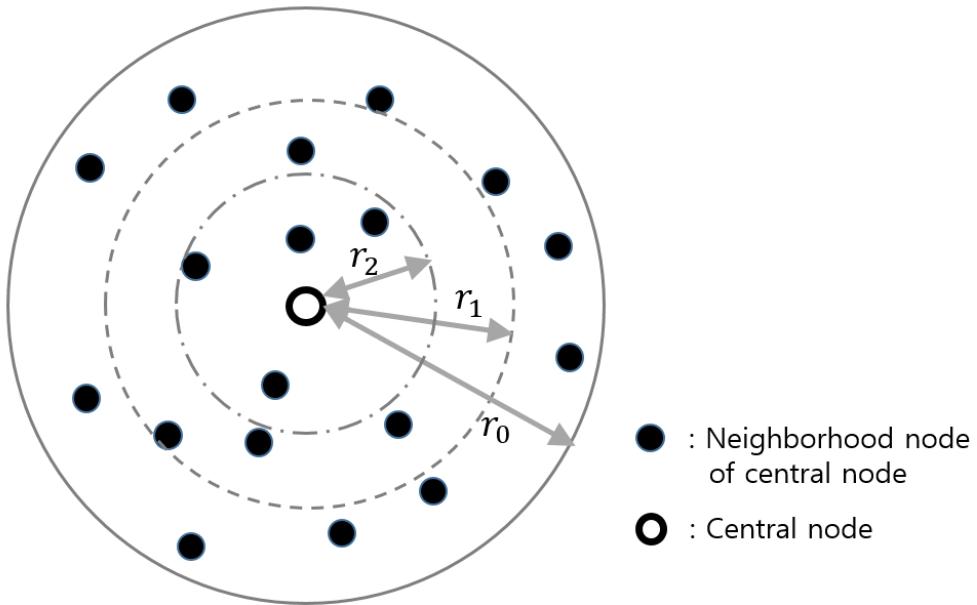


그림 4.2 DPC-MDS를 적용한 중심 노드의 커버리지 변화

DPC-MDS는 중심 노드의 정보량만을 고려한 알고리즘이다. 어떤 노드가 DPC-MDS를 통해 송신 전력을 낮춤으로써 연결되는 노드가 하나도 없어지게 되는 이웃 노드가 생길 수 있다. 따라서 DPC-MDS는 노드가 고르게 분포되어 있는 센서 네트워크 환경에서 적용되어야 한다. 고립되는 노드가 생길 경우 측위를 한 후에 노드 개수를 파악했을 때 실제 노드 개수보다 적은 노드 개수의 위치가 추정될 것이다.

제 2 절 Edge node Removed Multidimensional Scaling (ER-MDS)

DPC-MDS에서는 노드가 고르게 분포되어 있지 않으면 송신 전력 축소에 따라 고립되는 노드가 발생할 수도 있다. 이웃 노드의 고립 문제를 해결하기 위해 Edge node Removed Multidimensional Scaling(ER-MDS)에서는 가장 자리 노드의 연결 상태를 고려하여 송신 전력을 조절한다. 가장 자리 노드의 연결 상태가 우수하고 이웃 노드의 개수가 충분할 때, 중심 노드는 송신 전력을 줄인다.

$$\begin{aligned} Conn_Inf &= 1 \quad \text{where } neigh_i \geq NL \\ Conn_Inf &= 0 \quad \text{where } neigh_i < NL \end{aligned} \quad (4.2)$$

위의 식 (4.2)은 위치를 모르는 노드 n 개에 대하여 연결 상태를 정의하는 식이다. i 노드의 이웃 노드의 개수 $neigh_i$ 가 기준이 되는 이웃 노드의 개수 NL 이상일 경우 자신의 연결 상태가 좋은 것으로 인식하여 연결 정보(Connectivity Information, $Conn_Inf$)를 1로 정의한다. 그렇지 않은 경우엔 0으로 정의한다. 이웃 노드끼리 위와 같은 연결 정보를 공유하게 된다. 가장 자리 노드는 가장 약한 신호와의 거리를 통해 알 수 있다. 이 때, 가장 약한 신호를 보내는 노드의 연결 정보를 통해 송신 전력을 줄일 지 결정한다.

$$P_i = P_0 - 10 \cdot n \cdot \log_{10}(d_i/d_0) \quad (4.3)$$

식 (4.3)는 수신된 신호의 전력을 dBm 단위에서 계산하는 path-loss model이다. 모든 센서에는 신호를 인식하기 위한 최소 신호 세기가 존재한다. 이를 센서의 sensitivity라고 한다. 송신 전력을 줄임으로써 가장자리 노드를 이웃 노드에서 제거하려면 가장자리 노드에서 보낸 신호가 sensitivity 크기 아래로 떨어지면 된다.

$$\begin{aligned} P_{before} - P_{after} + (P_{sen} - P_{edge}) &= 0 \\ P_{after} &= P_{before} - (P_{edge} - P_{sen}) \end{aligned} \quad (4.4)$$

식 (4.4)는 ER-MDS에서 축소된 전력을 구하는 식이다. P_{before} 는 조절되기 전 센서의 송신 전력이고, P_{sen} 는 센서의 sensitivity 크기, P_{edge} 는 가장자리 노드에서 받은 신호의 세기를 의미한다. 위와 같은 과정이 DPC-MDS와 동일하게 매 iteration에 적용되어 수정된 송신 전력과 이웃 노드를 통해 위치를 추정한다.

ER-MDS에서도 DPC-MDS와 마찬가지로 기준이 되는 이웃 노드의 개수 이하로 떨어지지 않게 송신 전력을 조절한다. *Conn_Inf*의 값이 0이 되면 송신 전력을 조절하기 위한 알고리즘을 수행하지 않는다.

알고리즘을 적용했을 때, 가상의 센서 네트워크는 아래의 그림 4.3와 같다. r_1, r_2, r_3, r_4 는 ER-MDS가 적용됨에 따라 조절된 송신 전력에 의한 커버리지의 반지름이다. 송신 전력이 줄어들에 따라 가장자리의 노드와의 connection이 없어지는 것을 확인할 수 있다. 이처럼 가장자리 노드의 위치에 따라 전력을 연속적으로 조절하는 알고리즘이 ER-MDS이다.

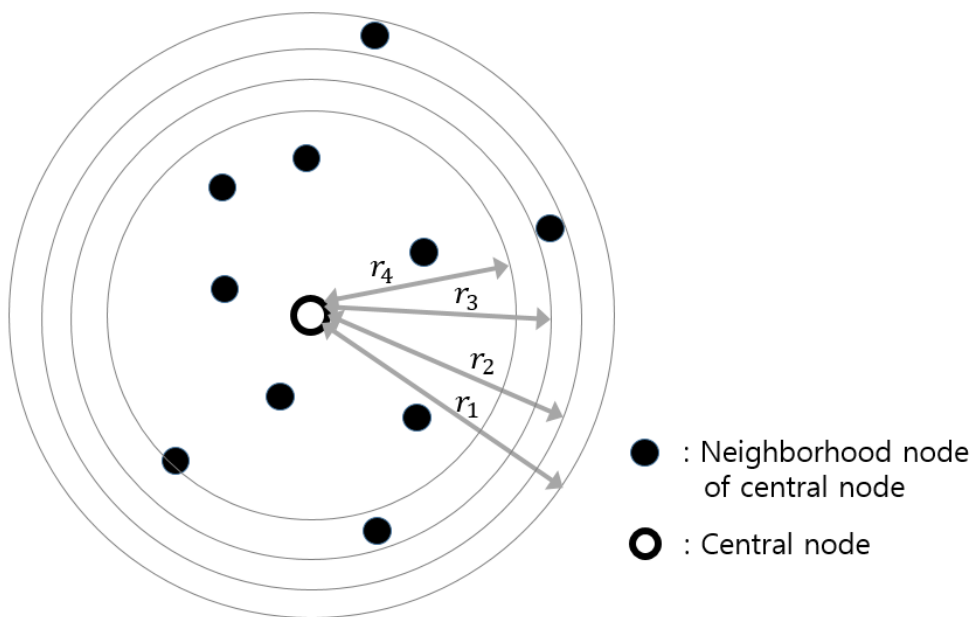


그림 4.3 ER-MDS를 적용한 중심 노드의 커버리지 변화

제 5 장 시뮬레이션 결과

본 장에서는 앞서 4 장에서 제시했던 두 알고리즘에 대한 성능을 검증한다. 3 장에서 제시했던 삼변측량법, CMDS, DW-MDS와 비교하여 제공 평균 제곱근 오차(Root Mean Square Error)를 확인한다. 또한 두 알고리즘을 분산처리 측위 알고리즘인 DW-MDS와 비교하여 에너지를 얼마나 효과적으로 줄였는 지 비교하여 본다. DPC-MDS와 ER-MDS는 송신 전력을 조절하는 기준이 되는 이웃 노드 개수에 따라 소비되는 에너지와 정확도가 달라지게 된다. 이에 대한 경향성을 본 장에서 확인해보도록 한다. 시뮬레이션 환경은 아래의 그림 5.1과 같이 40m X 40m 환경을 가정한다. 노드는 임의적으로 위치한다. 앵커 노드(Anchor node)는 위치를 알고 있다고 가정하는 노드이다. 앵커 노드는 그림 5.1의 다이아몬드 모양과 같이 시뮬레이션 환경의 가장 자리에 위치하고 있다.

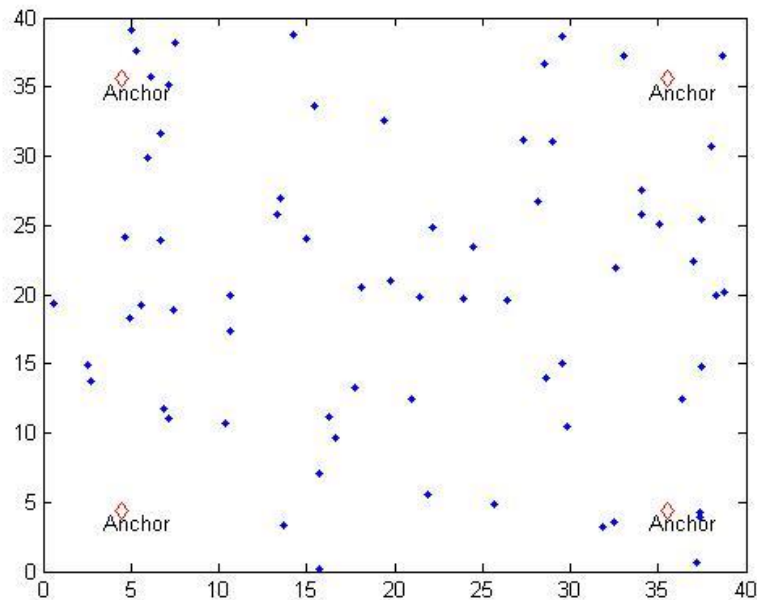


그림 5.1 40m × 40m 환경에서 앵커 노드와 타겟 노드

노드와 노드 사이의 거리는 RSS를 측정한 후에 Pathloss Model 식에 대입하여 거리를 추정한다. Pathloss exponent를 구해야만 RSS를 측정했을 때, RSS를 기반으로 거리를 측정할 수 있다. 대략적인

exponent 값을 알기 위해 15.4m X 22.6m 실내 Hall에서 CC2420 Micaz Radio Module을 이용하여 실험을 진행하였다. CC2420의 송신 전력은 0dBm, RSS sensitivity는 -90dBm이며, 2.4GHz 주파수를 사용한다. 본 실험에서는 중심 위치에 대해 0.5λ 씩 간격을 두어서 small scale fading을 줄였다. $d_0=1.2\text{m}$ 에서 P_0 는 -58dBm였으며, 이를 기준으로 하여 2.4m, 3.6m, 4.8m에서 LOS(Line-of-Sight) 환경의 RSS를 측정하였다. 실험 결과는 아래의 그림 5.2와 같다. pathloss exponent n_p 는 약 3.02의 값을 가졌으며, 이는 IEEE 802.15.4 Channel model에서 제시하고 있는 2~4 사이의 값을 만족한다. 또한 다양한 실험 연구에서도 실내 환경의 경우엔 2~3, 실외 환경의 경우엔 2~4 사이의 값을 가지는 것으로 알려져있다[12-13].

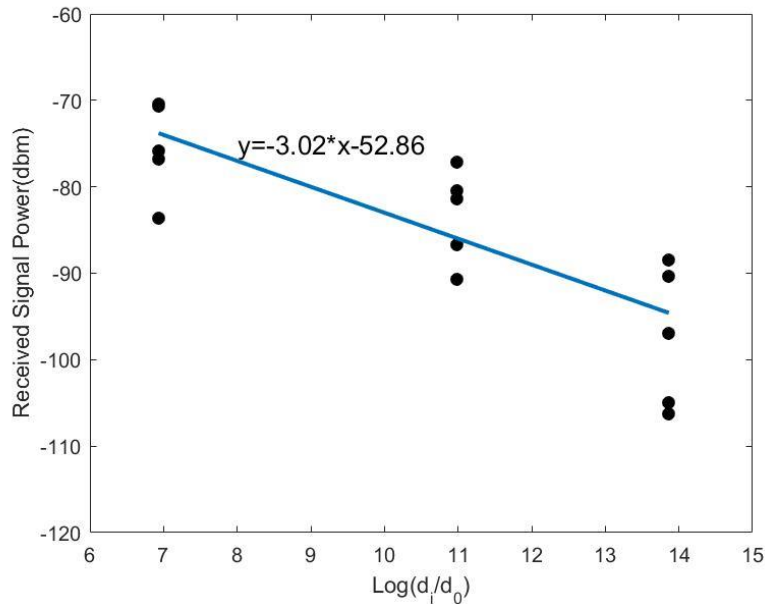


그림 5.2 15.4m x 22.6 실내에서의 Pathloss exponent

Path-loss exponent n_p 와 센서 노드의 sensitivity를 이용하여 커버리지의 최대거리를 구할 수 있다. Sensitivity란 노드가 신호를 인식할 수 있는 최소 전력의 크기를 의미한다. MOTE를 기준으로 sensitivity를 -90dBm, 1m에서의 path-loss를 -55dBm이라고 한다면 0dBm 송신 전력을 가지는 노드의 커버리지의 최대거리 d_{cov} 는 아래의 식처럼 14.68m이다.

$$\begin{aligned}
-90 &= -55 - 10 \cdot n_p \log_{10} d_{cov} \\
d_{cov} &= 10^{\frac{90-55}{10 \cdot n_p}} \simeq 14.68\text{m}
\end{aligned} \tag{5.1}$$

분산처리 측위에서는 커버리지 범위 안에 있는 노드의 정보만으로 측위하며, CMDS와 삼변측량에서는 멀티홉 통신을 통해서 모든 노드와의 거리를 계산하여 위치를 추정한다. 시뮬레이션마다 앵커 노드의 위치는 동일하며, 타겟 노드의 위치를 임의로 위치시킨다. 시뮬레이션 횟수는 10^3 으로 하였다.

RSS에 의한 i 노드와 j 노드 사이의 추정 거리는 아래와 같이 Gaussian noise model을 따른다.

$$\delta_{ij} = d_{ij}(1 + \mathcal{N}(0, \sigma)) \tag{5.2}$$

δ_{ij} 는 i 노드와 j 노드 사이의 추정거리이며, d_{ij} 는 실제 거리를 의미한다. $\mathcal{N}(0, \sigma)$ 는 평균이 0이고 분산이 σ 인 Gaussian random variable을 의미한다. 식 (5.2)는 1홉으로 통신이 가능한 노드 간의 추정 거리를 계산할 때의 오차를 의미한다. 다시 말해서, 이웃 노드 간의 추정 거리이다. CMDS와 삼변측량법에서는 각각 모든 노드 간의 거리와 타겟노드와 앵커 노드 간의 거리 정보가 필요하다. 1홉 이상으로 연결된 노드 간의 거리는 홉의 수를 이용하여 추정 거리를 계산할 수 있다.

$$\delta_{ij} = d_{cov} \cdot hop_{ij} \tag{5.3}$$

hop_{ij} 은 i 노드와 j 노드 사이의 최소 홉의 개수를 의미한다. 홉의 개수는 Dijkstra 알고리즘을 통해서 구할 수 있다. 멀티홉으로 연결된 노드 간의 거리는 단순히 홉의 수와 거리의 곱으로 나타내기 때문에 추정 거리의 오차가 1홉으로 통신되는 노드 사이의 추정 거리보다 크다.

제 1 절 거리 오차를 통한 알고리즘 성능 분석

거리 오차는 모든 알고리즘에 대해 추정된 위치와 실제 위치의 차이의 제곱 평균의 제곱근(Root Mean Square Error, RMSE)을 이용한다. RMSE는 아래의 식 (5.4)와 같이 구한다.

$$\text{RMSE} = \frac{1}{N} \sum_{i=1}^N \sqrt{(p_i - \tilde{p}_i)^T (p_i - \tilde{p}_i)} \quad (5.4)$$

여기서, p_i 는 i 노드의 위치 벡터($= (x_i, y_i)^T$), \tilde{p}_i 는 i 노드의 추정된 위치 벡터($= (\hat{x}_i, \hat{y}_i)^T$)를 의미한다.

본 논문에서 제시한 분산처리 측위에서는 매 iteration마다 더 정확한 위치를 찾아나간다. 그만큼 iteration 횟수 또한 알고리즘의 성능에 영향을 미치게 된다. Iteration 수가 많을수록 정확한 위치를 찾게 되지만 그만큼 localization period, communication period가 길어져 비효율적인 시스템이 된다. 본 논문에서는 RMSE 거리 3m가 보장되는 범위 안에서 iteration을 수행하였으며, 이를 위해서 8회 혹은 12회에 대해서 시뮬레이션 하였다. localization time을 Iteration 횟수에 따른 정확도의 변화는 아래와 같다.

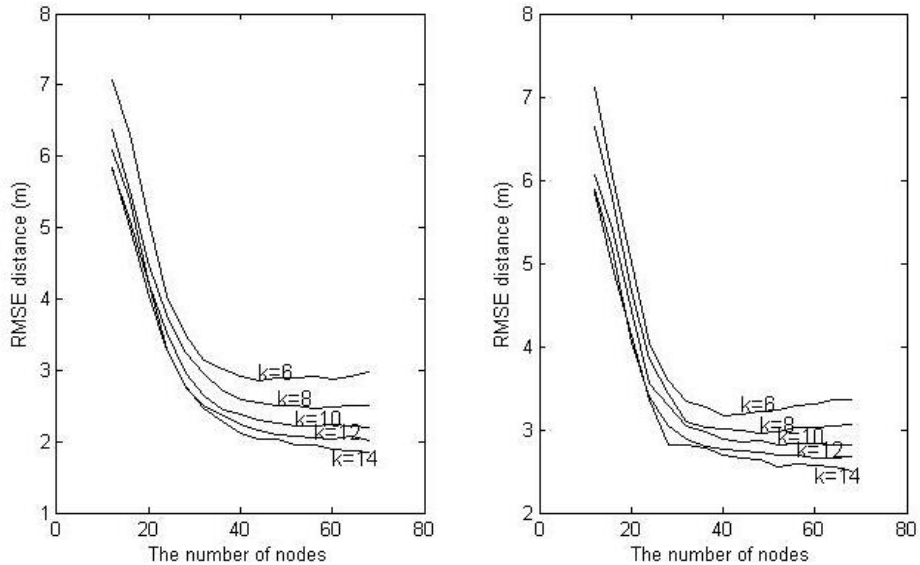


그림 5.3 iteration 횟수에 따른 ER-MDS(왼쪽)와 DPC-MDS(오른쪽)의 RMSE

아래의 그림 5.3은 식 (5.4)을 토대로 NL(Node Limit)=8, iteration 횟수는 12회, 측정오차의 variance $\sigma = 0.1$ 일 때 노드의 개수를 12개부터 64개에 대하여 모든 알고리즘에서 RMSE를 분석한 그래프이다.

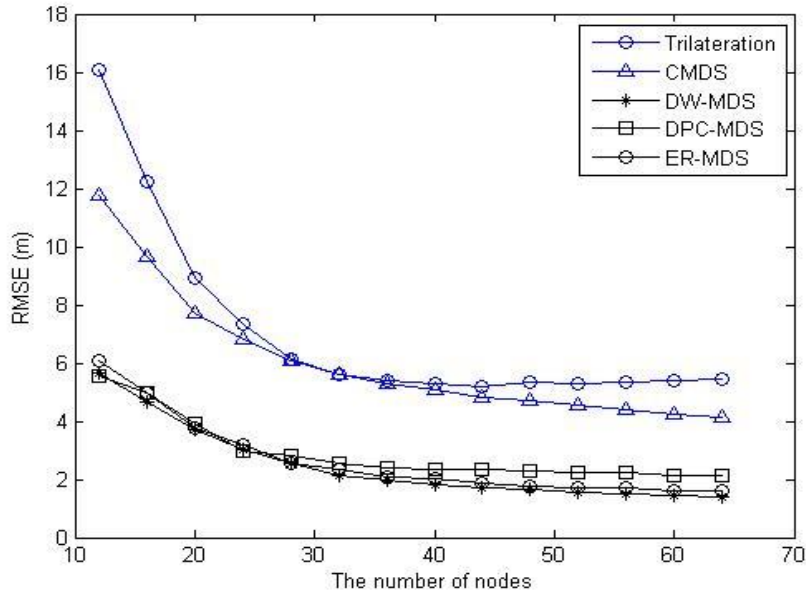


그림 5.4 알고리즘 별 노드 개수에 따른 RMSE

위의 그래프에서 알 수 있듯이 5개의 알고리즘 모두 노드의 개수가 증가할수록 RMSE가 줄어드는 것을 확인할 수 있다. MDS를 이용한 알고리즘에서 특히 이러한 경향을 잘 보여주는데, CMDS의 경우에도 상대적으로 비교할 정보량이 많아지기 때문에 정확도가 높아지며, DW-MDS도 노드의 개수가 증가하면 이웃 노드의 개수도 증가하기 때문에 중심 노드의 추정 위치와 비교할 노드가 많아져서 iteration이 진행함에 따라 정확한 위치로 업데이트하게 된다. DPC-MDS와 ER-MDS도 마찬가지로 이유로 정확도가 높아진다.

제안한 두 알고리즘과 DW-MDS를 비교하면 DW-MDS가 두 알고리즘에 비해 더 높은 정확도를 보인다. 세 알고리즘 모두 분산처리 측위 알고리즘으로써 이웃 노드의 개수가 많을수록 더 많은 정보량을 이용해 높은 정확도로 위치를 추정할 수 있다. 그런데 제안한 두 알고리즘은 송신 전력을 줄임으로써 이웃 노드의 개수를 줄이거나 유지하도록 하고 있다. 다시 말해서, DW-MDS보다 적은 개수의 이웃 노드로 인해 RMSE

가 증가한 것이다.

아래의 그림은 40m x 40m 시뮬레이션 환경에서 임의적으로 분포된 노드의 추정 위치를 분포로 표현한 그림이다. Iteration 횟수는 12번, 위치를 모르는 노드의 개수는 40개, 거리 오차의 분산은 0.1을 가지는 환경에 대하여 알고리즘을 적용하였다.

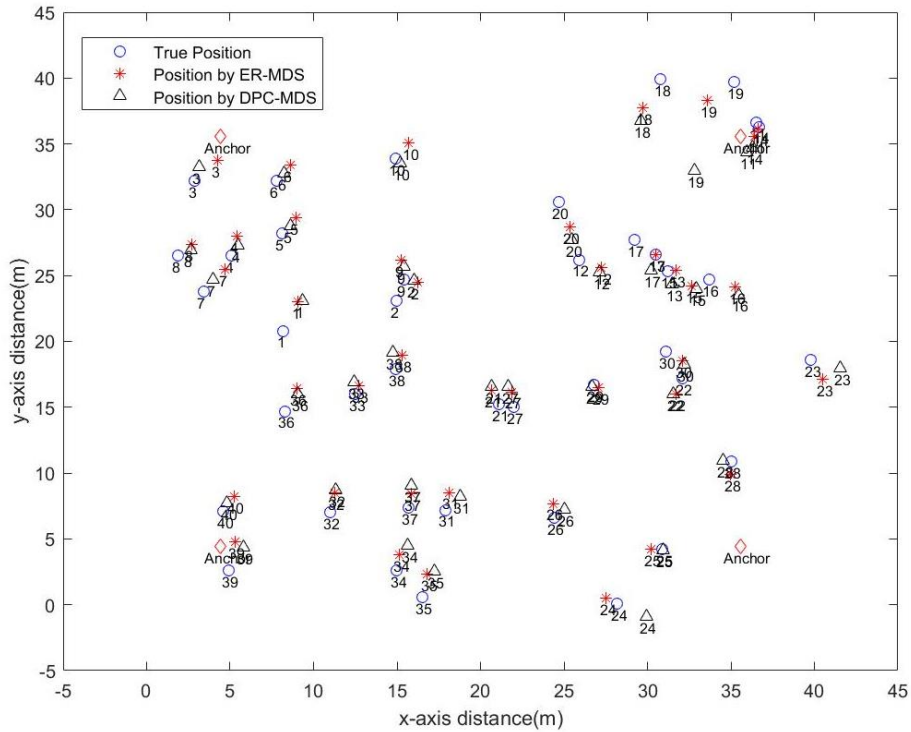


그림 5.5 알고리즘 결과 추정된 노드의 위치

위의 그림에서 파란색 원은 실제 위치, 빨간색 별은 ER-MDS에 의해 추정된 노드의 위치, 검정색 삼각형은 DPC-MDS에 의해 추정된 노드의 위치를 의미한다. DPC-MDS에 의해 추정된 위치보다 ER-MDS에 의해 추정된 위치가 비교적 정확한 것을 알 수 있다.

표 5.1 오차 variance와 노드 개수에 따른 RMSE 비교

Distance error variance	The number of nodes	DPC-MDS	ER-MDS	DW-MDS

$\sigma = 0.1$	16	4.79m	4.86m	4.63m
	32	2.52m	2.38m	2.17m
	64	2.21m	1.61m	1.40m
$\sigma = 0.15$	16	4.82m	4.88m	4.69m
	32	3.21m	2.90m	2.73m
	64	2.83m	2.31m	2.01m
$\sigma = 0.2$	16	5.03m	5.02m	4.92m
	32	4.22m	3.96m	3.75m
	64	4.15m	3.77m	3.22m

위의 표는 DW-MDS, DPC-MDS, ER-MDS 알고리즘의 RMSE를 노드 개수와 error variance에 따라 수치화한 표이다. Iteration 횟수는 12회로 고정하였다. 실제 노드 사이의 거리 d_{ij} 와 RSS로 측정한 노드 사이의 거리 δ_{ij} 의 오차 variance가 증가할수록 측위 알고리즘의 오차도 증가하게 된다. Variance가 0.1일 때는 노드 개수에 따른 RMSE의 감소가 잘 나타나는 반면, variance가 증가할수록 노드 개수가 증가하더라도 눈에 띄는 감소가 나타나지 않는 것을 알 수 있다. MDS를 이용한 분산 처리 측위가 이웃 노드와의 거리 정보를 기반으로 한 측위이기 때문에 많은 iteration을 통해 위치를 추정하더라도 측정 거리 오차보다 더 적은 오차로 위치를 추정하기 힘들 것이다.

제 2 절 DW-MDS와 제안한 알고리즘의 에너지 소모 비교

측위 과정에서 생기는 에너지 소모는 2가지 근본적인 이유가 있다. 첫 번째로 신호를 송수신할 때 쓰이는 에너지, 두 번째는 알고리즘을 프로세싱하는 데 쓰이는 에너지로 나뉘 수 있다. RSS 신호를 기반으로 한 위치 측위에서는 노드 사이에 얼마나 많은 bit를 이용해서 정보를 송수신하는 지, 얼마나 자주 정보를 주고 받는 지에 따라 에너지의 소모가 결정된다. 본 논문에서 에너지 소모량을 분석할 때엔 전체 MAC protocol에 따르는 에너지 소모량이 아닌 측위 과정에서 소모된 에너지만을 분석하였다. 소모된 에너지를 계산하는 식은 아래의 식과 같은 모델을 적용한다.[15]

$$E_i = E_{proc,i} + \sum_{j=1}^{iter} (P_{i,Tx}^{(j)} \cdot t_{i,Tx}^{(j)} + P_{i,Rx}^{(j)} \cdot t_{i,Rx}^{(j)}) \quad (5.5)$$

식 (5.5)에서 E_i 는 i 노드에서 한번의 측위 알고리즘에서 소모된 에너지를 의미한다. $E_{proc,i}$ 는 i 노드에서 측위 알고리즘을 수행하는 데 쓰이는 에너지이다. $P_{i,Tx}^{(j)}$ 와 $P_{i,Rx}^{(j)}$ 는 i 노드에서 j 번째 iteration에서 소모된 송신 전력과 수신 전력을 의미한다. $t_{i,Tx}^{(j)}$ 와 $t_{i,Rx}^{(j)}$ 는 i 노드에서 송신 신호의 Transmisstion 시간과 수신 신호를 받는 시간을 의미한다. $E_{proc,i}$ 는 신호를 송수신하는데 쓰이는 에너지에 비해서 매우 작으므로, 본 논문에서는 송수신 시 사용되는 에너지만을 모델링하여 비교한다. 송신 전력은 노드에서 어떤 전력으로 데이터를 보낼 것인지에 따라 달라지지만 수신 전력은 일정한 크기를 가지고 있다. 아래의 표는 CC2420의 송수신 전력에 관한 표이다[14].

표 5.2 송신 모드에서의 소모 전력 및 에너지

$P_{out}[dbm]$	$I_{Tx}[mA]$	$P_{Tx}[mW]$	$E_{Tx-1packet}[\mu J]$
0	17.4	31.32	40.09
-1	16.5	29.70	38.02
-3	15.2	27.36	35.02
-5	13.9	25.02	32.02
-7	12.5	22.50	28.80
-10	11.2	20.16	25.80
-15	9.9	17.82	22.81
-25	8.5	15.30	19.58

표에서 P_{out} 은 radio output에서의 signal power level을 의미하며, P_{Tx} 가 송신 단계에서 소모된 전력이다. Signal level에 따라 소모하는 전류의 크기가 다르며, 전류의 크기에 supplied voltage인 1.8V를 곱해주어서 P_{Tx} 를 구한다. 1packet당 소모에너지는 1개의 packet 길이를 320bit라고 하였을 때, CC2420의 data rate는 250kbps이므로 $320/250=1.28ms$ 가 나온다. 이 시간을 P_{Tx} 에 곱해주어 1 packet당 에

너지를 구할 수 있다. 마찬가지로 1 packet당 수신 에너지도 구할 수 있다. 아래의 표 5.3은 수신 모드에서 소모되는 에너지를 제시한 표이다.

표 5.3 수신 모드에서의 소모 전력 및 에너지

$I_{Tx} [mA]$	$P_{Tx} [mW]$	$E_{Tx-1packet} [\mu J]$
18.8	33.84	43.31

위의 표 5.2와 5.3에 제시된 송수신 전력을 적용하여 알고리즘 별로 소모된 에너지를 계산하였다. 수신 에너지의 경우 1홉으로 통신할 수 있는 이웃 노드의 개수에 따라 선형적으로 증가하게 된다. DPC-MDS에서는 signal power level에 따른 전력을 이용하여 소모된 에너지를 계산하였다. $TP_0 = 0 \text{ dbm}$, $TP_1 = -3 \text{ dbm}$, $TP_2 = -7 \text{ dbm}$ 의 signal power level을 적용하였다. ER-MDS에서는 표 5.2를 이용하여 사이 값을 interpolation하여 에너지를 계산하였다. 식 (5.5)와 표 5.2, 표 5.3을 적용하여 계산한 전체 에너지 소모량은 아래의 그림 5.6과 같은 추이를 보였다.

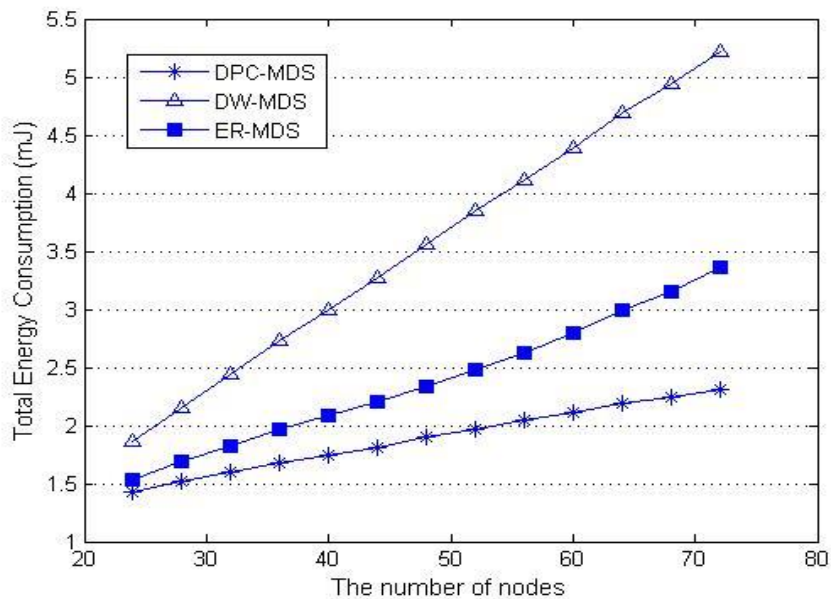


그림 5.6 각각의 알고리즘에서 소모되는 에너지

위의 시뮬레이션은 iteration=8, NL=8, 거리 오차 분산이 0.1일 때

시뮬레이션을 실행한 결과이다. 위의 그래프에서 알 수 있듯이 DPC-MDS와 ER-MDS 모두 기존의 DW-MDS에 비해서 에너지 소모량이 줄어든 것을 알 수 있다. DPC-MDS가 ER-MDS보다 더 적은 에너지가 소모되는 이유는 DPC-MDS의 경우 이웃 노드의 개수를 파악한 후에 signal power level을 낮춤으로써 한꺼번에 많은 전력을 줄이지만 ER-MDS의 경우에는 서서히 최적화하기 때문에 변화폭이 적다. DPC-MDS는 DW-MDS에 비해서 55% 정도의 에너지 절약 효과를 보였으며, ER-MDS는 DW-MDS에 비해서 약 35% 정도의 에너지 절약 효과를 가져왔다. 제안한 두 알고리즘에서 에너지 절약 효과를 보인 또 다른 이유는 송신 전력을 줄임으로써 이웃 노드의 개수도 줄었기 때문이다. 이웃 노드의 개수를 줄임으로써 신호를 수신하는 데 쓰이는 에너지가 줄어들게 되었다.

DPC-MDS와 ER-MDS 모두 노드의 송신 전력을 조절하는 알고리즘이다. 각 알고리즘에서의 노드 1개에 대한 평균 송신 에너지는 다음과 같다.

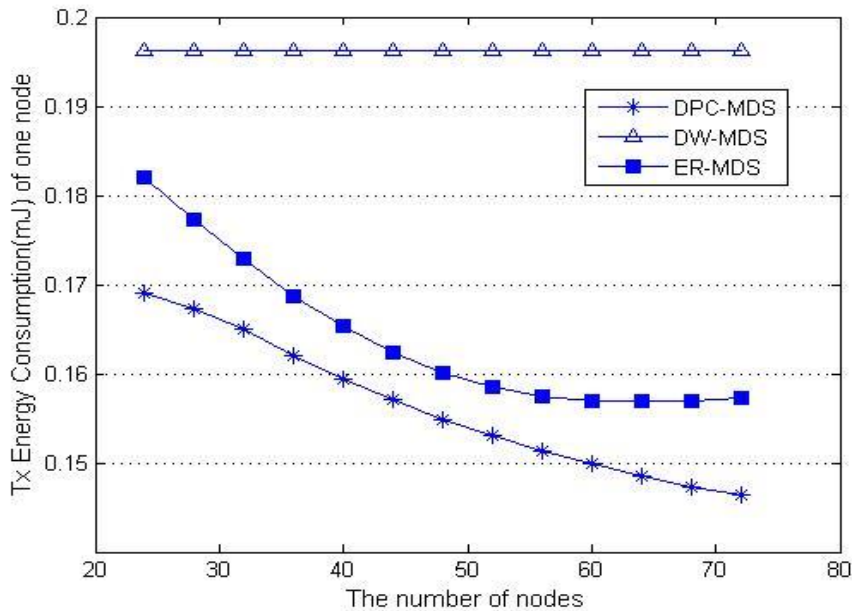


그림 5.7 노드 1개에 대한 평균 송신 에너지

DW-MDS는 0 dBm의 signal power level을 유지하기 때문에 노드 개수에 상관없이 항상 일정한 값을 유지하고 있다. ER-MDS와 DPC-MDS는 모두 노드의 개수가 증가함에 따라 평균 송신 에너지가 줄어드

는 것을 알 수 있다. 이것은 노드의 밀도가 증가하면서 이웃 노드도 함께 증가하기 때문에 점점 작은 에너지로도 많은 이웃 노드를 확보할 수 있기 때문이다. 그리고 노드의 개수가 많아지면서 감소폭이 줄어들다가 점점 saturation 되는 경향을 보인다. 이는 iteration 수가 고정되면서 줄일 수 있는 에너지의 양이 한정되어 있기 때문인 것으로 해석할 수 있다. 만약 충분한 측위 시간이 보장된다면 iteration 횟수를 늘림으로써 더 적은 에너지로 위치 추정을 할 수 있게 될 것이다.

제 3 절 기준 이웃 노드 개수에 따른 알고리즘 성능 변화

제안한 두 알고리즘에서는 송신 전력을 조절하기 위한 노드 개수의 기준에 따라 RMSE 거리와 소비되는 에너지의 크기가 달라진다. 예를 들어, 같은 노드 밀도를 가진 환경에서 DPC-MDS의 경우 송신 전력을 줄이기 위한 노드의 개수가 5개일 때는 10개일 때보다 더 쉽게 송신 전력을 줄일 수 있으며, 이에 따라 에너지는 줄일 수 있어도 정확도는 낮아질 수 있다. ER-MDS 또한 노드 개수를 최소한 몇 개를 보장할 것인가에 따라 RMSE와 소비된 energy가 달라질 수 있다. 아래의 그림은 기준 이웃 노드 개수를 다르게 한 DPC-MDS와 ER-MDS의 RMSE 성능을 비교한 그래프이다.

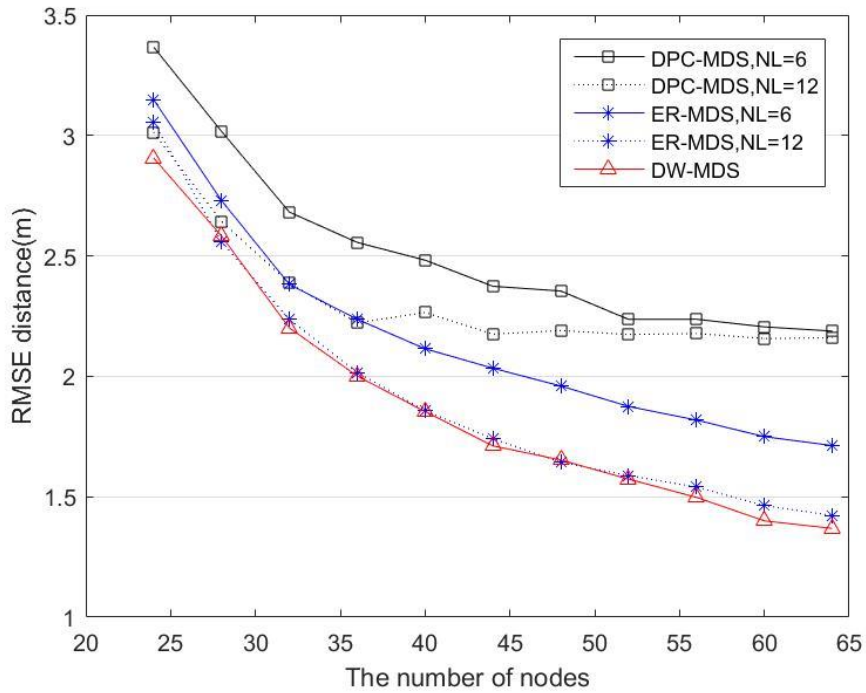


그림 5.8 이웃 노드 개수의 제한에 따른 DPC-MDS와 ER-MDS의 RMSE 성능 비교

NL이 6일 때와 12일 때, RMSE 거리를 비교하였다. NL이 작아질수록 RMSE 성능이 나빠지는 것을 확인할 수 있다. 이것은 NL이 작아지면서 각 노드들이 더 작은 에너지로 송수신하기 때문이다. MDS를 이용한 분산 측위에서는 이웃 노드들과의 거리를 통해 자신의 위치를 찾아나가는 데 이웃 노드의 정보들이 줄어들어서 자신의 위치를 더 부정확하게 찾게 되는 것이다. ER-MDS가 DPC-MDS에 비해서 성능이 좋은 이유는 ER-MDS의 경우 가장 자리에 있는 노드의 연결 상태에 따라 서서히 송신 전력을 조절하기 때문에 각 iteration에서 DPC-MDS보다 더 많은 거리 정보를 이용하게 된다.

NL이 클수록 정확한 위치 추정이 가능한 대신에 더 많은 에너지 소모를 필요로 한다. NL을 많이 확보할수록 더 많은 수신 신호를 받게 되고, 받은 packet에 따라 비례하게 증가하는 수신 에너지 때문에 노드의 에너지 소모가 증가한다. 아래의 그림은 NL이 6과 12일 때, 각각에 대한 에너지 소모량을 비교한 그래프이다.

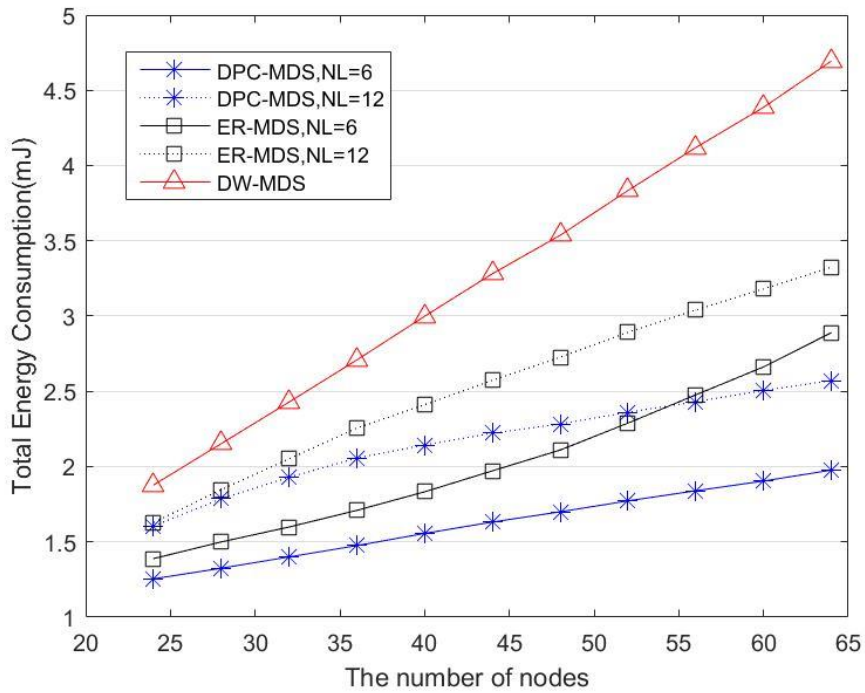


그림 5.9 이웃 노드 개수의 제한에 따른 알고리즘의 에너지 소모량 비교

위의 시뮬레이션은 iteration=8, 거리 오차 분산이 0.1일 때 시뮬레이션을 실행한 결과이다. 위의 그림에서 알 수 있듯이 ER-MDS와 DPC-MDS 모두 NL이 12일 때가 6일 때보다 더 많은 에너지를 소모하고 있다. DW-MDS는 노드를 감지하는 과정이 없기 때문에 하나의 그래프로 표현하고 있다. DPC-MDS의 경우 NL이 6일 때는 DW-MDS에 비해서 약 60%의 에너지를 줄이고, NL=12일 때는 약 48%의 에너지를 줄인다. ER-MDS의 경우, NL=6일 때는 DW-MDS에 비해서 약 37%의 에너지를 줄이며, NL=12일 때는 약 30%의 에너지를 줄인다. 이러한 경향은 노드 1개에서의 평균 송신 에너지에서도 찾아볼 수 있다.

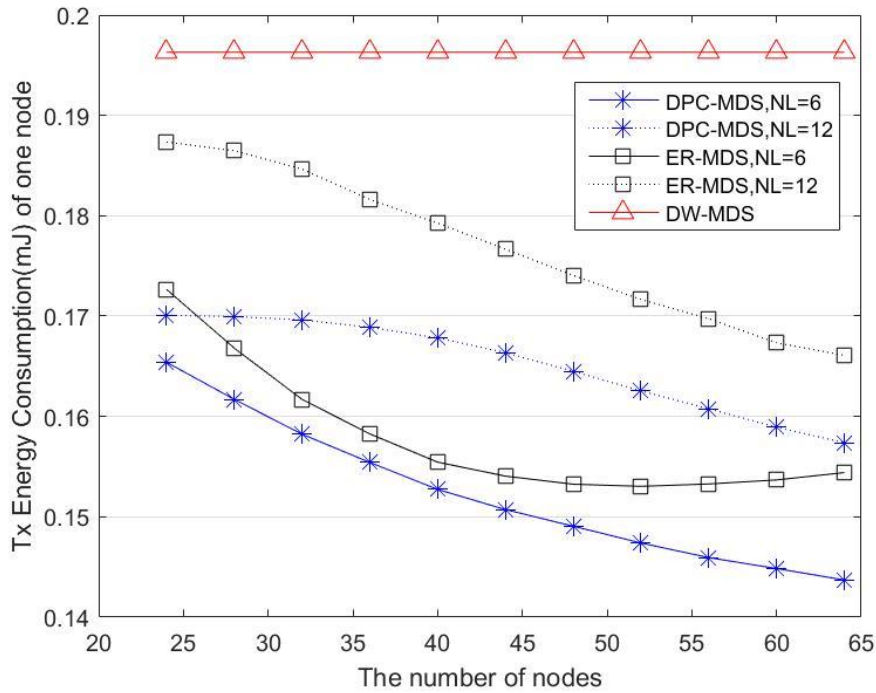


그림 5.10 이웃 노드 개수 제한이 6, 12일 때 전체 노드 개수에 따른 노드 1개의 평균 송신 에너지

DW-MDS에서는 일정한 송신 전력을 유지하기 때문에 변화가 없다. NL이 6일 때, DPC-MDS는 NL이 12인 경우에 비해서 노드의 개수에 따라 에너지가 줄어드는 폭이 크다. 그만큼 평균적으로 노드들이 쉽게 송신 전력을 줄일 수 있다는 것을 의미한다. NL이 6이면 DPC-MDS의 송신 에너지는 DW-MDS에 비해서 약 26% 정도 감소되며, NL이 12이면 약 18% 감소된다. ER-MDS는 가장 자리의 노드의 연결 상태에 따라 차례로 전력을 줄이기 때문에 감소 폭은 NL이 6일 때와 12일 때가 동일하다. 다만 노드의 밀도가 높아질수록 모든 노드가 iteration이 끝났을 때 동일한 전력을 가질 수 있다. 모든 iteration에서 전력을 줄였기 때문에 특정 노드 밀도를 넘어서선 노드 1개에 대한 평균 송신 전력이 일정해진다. NL이 6일 때, ER-MDS는 DW-MDS에 비해서 약 22% 정도의 에너지 감소를 보이며, NL이 12일 때는 약 15%의 감소를 보인다.

제 4 절 제안한 알고리즘과 DW-MDS의 복잡도 비교

제안한 두 알고리즘과 DW-MDS는 모두 이웃 노드의 개수가 알고리즘의 속도에 영향을 미친다. 식 (3.21)에서도 알 수 있듯이 매 iteration에서 위치를 찾는 식이 이웃 노드의 개수에 비례하여 복잡도가 증가한다. 이웃 노드의 개수가 m 개라고 한다면 세 알고리즘 모두 a_i 와 $\{b_j^{(k-1)}\}_{j=1,2,\dots,N}$ 을 구하는 과정에 의해 알고리즘의 복잡도는 $O(2m)$ 이 된다. 각 알고리즘에 대한 이웃 노드의 개수의 변화는 아래의 그림과 같다.

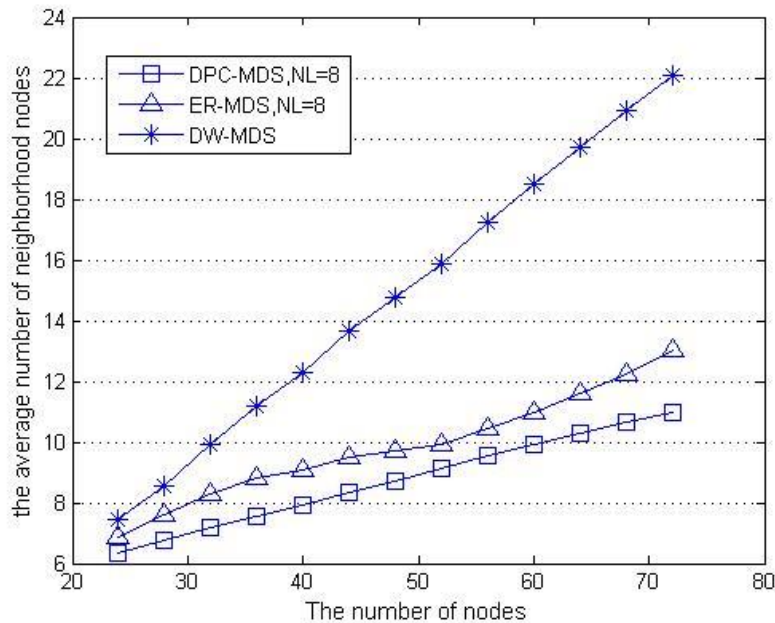


그림 5.11 NL=8일 때, 이웃 노드 개수의 변화

위 그림은 이웃 노드 제한(NL)을 8개로 두고 DPC-MDS와 ER-MDS를 실행하고, 이를 DW-MDS와 비교한 그래프이다. DPC-MDS는 이산적으로 전력을 조절하기 때문에 power level에 따른 이웃 노드의 감소가 크다. ER-MDS는 가장 자리의 노드의 connectivity를 고려하여 연속적으로 줄이기 때문에 이웃 노드의 감소가 DPC-MDS에 비해선 적다. 이웃 노드 개수를 몇 개로 제한할 것인가도 알고리즘의 Complexity에 영향을 미친다. NL이 크면 많은 이웃 노드를 보장하려고 하기 때문에 송신 전력을 조금만 줄이게 되고, Complexity도 작은 NL에 비해서 클

것이다. 아래의 그림은 NL이 8일 때와 12일 때의 DPC-MDS와 ER-MDS의 이웃 노드 개수를 그린 그래프이다.

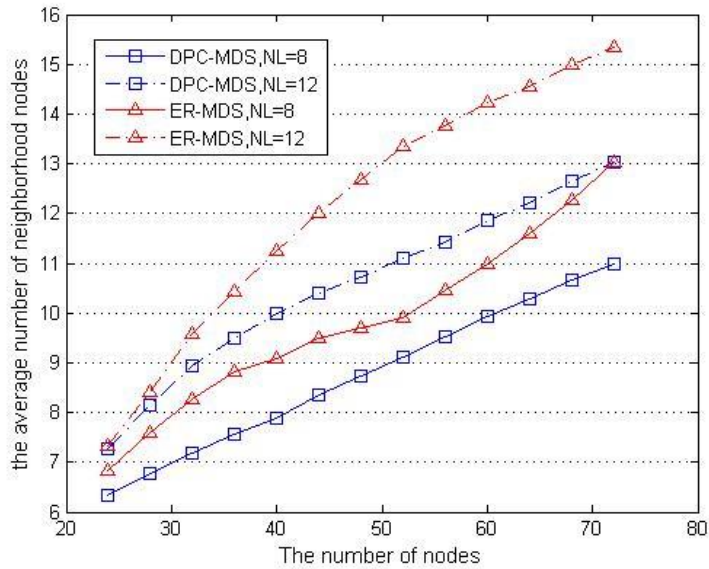


그림 5.12 NL=8,12일 때 DPC-MDS와 ER-MDS에서의 이웃 노드 개수

위의 시뮬레이션은 iteration이 12, 측정 오차 $\sigma = 0.1$ 일 때의 결과이다. 측정 오차가 달라져도 노드의 밀도가 달라지는 것은 아니므로 비슷한 결과를 낼 것이다. 위의 그래프를 통해서 NL이 클수록 더 많은 이웃 노드를 보장하는 것을 알 수 있다. NL이 작을수록 작은 Complexity를 가지고 적은 에너지가 소모되지만 정확도에서 열화를 보이게 된다.

제 6 장 결론

본 논문에서는 무선 센서 네트워크에서 에너지를 효율적으로 사용하면서 위치 추정을 할 수 있는 알고리즘을 제안하였다. DPC-MDS와 ER-MDS 두 가지 알고리즘을 삼변측량법, CMDS, DW-MDS와 측위의 정확도와 에너지 소모 측면에서 비교 및 검증하였다. DPC-MDS는 이웃 노드의 개수에 따라 이산적으로 송신 전력을 조절하는 다차원 척도법이다. 이때, 기준이 되는 노드의 개수 이하로 떨어지지 않도록 송신 전력을 조절하게 된다. ER-MDS는 매 iteration 마다 가장 자리의 노드의 연결 상태를 확인하여 가장 자리의 노드의 연결상태가 좋으면 송신 전력을 줄여서 가장 자리에 거리 정보를 보내지 않는다. 이처럼 가장 자리 노드의 위치에 따라 송신 전력을 연속적으로 조절하는 방법을 ER-MDS라 한다.

두 가지 알고리즘의 시뮬레이션 결과, 중앙처리식 측위인 삼변측량법과 CMDS와 비교해보면 측위 오차도 작고, 분산처리를 통해 빠르게 측위할 수 있다는 장점을 가지고 있다. 분산 처리식 측위인 DW-MDS와 비교해보면 정확도 측면에서는 높은 송신 전력을 이용해 많은 이웃 노드를 사용하는 DW-MDS가 DPC-MDS와 ER-MDS보다 더 높은 정확도를 보였다. 제안한 알고리즘보다 더 많은 이웃 노드와의 상대적 거리를 이용하기 때문에 그만큼 정확한 측위가 가능하다. 하지만 정확도의 차이는 시뮬레이션 사이즈의 3% 이내의 오차이기 때문에 복잡도와 에너지에서 오는 이점을 생각한다면 충분히 허용 가능한 수준으로 볼 수 있다.

에너지 측면에서는 DPC-MDS에서는 40% 이상, ER-MDS는 30% 이상의 에너지를 절약할 수 있었다. 그 이유는 노드에서 사용되는 송신 에너지도 줄었으며, 이웃 노드로부터 신호를 받을 때 생기는 수신 에너지 또한 줄일 수 있었기 때문이다. 노드의 송신 전력이 줄어들음에 따라 커버리지가 좁아지면서 이웃 노드의 개수가 줄어들었다. 이에 비례하게 수신 에너지도 줄어든 것이다. 얼만큼 에너지를 줄일 것인가에 대한 것은 기준이 되는 이웃 노드의 개수를 어떻게 설정하느냐에 따라 달라진다. 개수를 매우 적게 설정하면 에너지는 많이 절약할 수 있지만 커버리지가 작아짐에 따라 이웃 노드의 개수가 적어지고, 측위의 정확도가 낮아진다.

알고리즘의 복잡도는 이웃 노드의 개수에 비례하는데, 커버리지가 좁아지면서 이웃 노드의 개수가 적어졌기 때문에 ER-MDS와 DPC-MDS 모두 DW-MDS에 비해서 복잡도가 줄어든 것을 확인할 수 있었다.

본 논문에서 제시한 알고리즘을 MAC protocol에도 적용한다면 보다 많은 에너지를 줄일 수 있을 것이다. 이후 연구에서는 이러한 네트워크 특성에 따른 알고리즘 연구를 진행할 수 있을 것이다.

참고 문헌

- [1] D. Kandris et al. “Power Conservation through Energy Efficient Routing in Wireless Sensor Networks”, *Sensors*, Sep, 2009, pp. 7320–7342.
- [2] J. Bachrach and C. Taylor, “Localization in Sensor Networks”, Cambridge, 2005.
- [3] T. Panda, N.K. Kamila and R.R. Patra, “Energy Efficient Anchor-Based Localization Algorithm for WSN”, *IOSR Journal of Computer Engineering*, Vol. 1, No. 3, May, 2012, pp. 13–20.
- [4] P. Tarrio, M. Cesana and A. Redondi, “Energy–accuracy trade-offs for hybrid localization using RSS and inertial measurements in wireless sensor networks”, *Ad Hoc Networks*, Vol. 11, Issue 6, Aug, 2013, pp. 1874–1889.
- [5] X. Shen, Z. Wang, P. Jiang, R. Lin and Y. Sun, “Connectivity and RSSI Based Localization Scheme for Wireless Sensor Networks”, *International Conference on Intelligent Computing*, Vol. 2, pp–23–26, 2005.
- [6] J. A. Costa, N. Patwari and A. O. Hero, “Distributed Weighted-Multidimensional Scaling for Node Localization in Sensor Networks”, *ACM Transactions on Sensor Networks*, Vol. 2, No. 1, pp. 39–64, Feb, 2006.
- [7] Yi Shang and W. Ruml, “Improved MDS-based Localization”, *IEEE INFOCOM*, Vol. 4, pp. 2640–2651, March, 2004.
- [8] L. M. Kaplan, “Local node Selection for Localization in a Distributed Sensor Network”, *IEEE Transactions on Aerospace and Electronic System*, Vol. 42, No. 1, Jan, 2006, pp 136–142.

- [9] S. Lederer, Y. Wang and J. Gao, “Connectivity–based Localization of Large Scale Sensor Networks with Complex Shape”, *ACM Transactions on Sensor Networks*, Vol. 5, No. 4, Nov, 2009, pp. 31:1–31:32.
- [10] E.W. Dijkstra, “A note on Two Problems in Connexion with Graphs”, *Numerische Mathematik*, 1, pp. 269–271, 1959.
- [11] E. Kim, S. Lee, C. Kim and K. Kim, “Mobile Beacon–based 3D–Localization with Multidimensional Scaling in Large Sensor Networks”, *IEEE Comm. Letters*, Vol. 14, No. 7, July, 2010, pp. 647–650.
- [12] O. Musikanon and W. Chongburee, “Zigbee Propagations and Performance Analysis in Last Mile Network”, *International Journal of Innovation, Management and Technology*, Vol. 3, No. 4, Aug, 2012, pp 353–357.
- [13] A. F. Molisch et al, “IEEE 802. 15. 4a Channel Model – final report”, <http://www.ieee802.org/15/pub/TG4a.html>, Online Link
- [14] “CC2420 Single–Chip 2.4GHz IEEE 802.15.4 Compliant and Zigbee Ready RF Transceiver”, <http://www.ti.com>, Online Link.
- [15] A. Barberis, L. Barboni and M. Valle, “Evaluating Energy Consumption in Wireless Sensor Networks Application”, *Euromicro Conference on Digital System Design Architectures*, pp. 455–462, Aug, 2007.

Appendix. A.

A.1 SMACOF algorithm

식 (3.20)의 i 노드에 대한 부분 비용함수 S_i 를 전개하면 아래의 식 (A.1)과 같이 표현할 수 있다.

$$\begin{aligned}
 S_i(p_i) &= \eta_\delta^2 + \eta^2(P) - 2\rho(P) \\
 \eta_\delta^2 &= \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} \delta_{ij}^2 + \sum_{j=n+1}^{n+m} 2w_{ij} \delta_{ij}^2 \\
 \eta^2 &= \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} d_{ij}^2(P) + \sum_{j=n+1}^{n+m} 2w_{ij} d_{ij}^2(P) \\
 \rho(P) &= \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} \delta_{ij} d_{ij}(P) + \sum_{j=n+1}^{n+m} 2w_{ij} \delta_{ij} d_{ij}(P)
 \end{aligned} \tag{A.1}$$

식 (A.1)의 η_δ^2 는 I 노드의 위치 벡터 p_i 와 관계없는 수식이며, $\rho(P)$ 는 p_i 에 비례하고, η^2 은 제곱에 비례한다. S_i 를 최소화하는 p_i 를 구하기 위해 majorizing 함수 T_i 를 정의할 수 있다. Majorizing 함수란 아래의 2가지 조건을 만족하면서 순차적으로 어떤 목적함수의 최소값을 찾아주는 함수를 의미한다.

$$\begin{aligned}
 (1) \quad S_i(\mathbf{p}) &\leq T_i(\mathbf{p}, \mathbf{q}) \\
 (2) \quad S_i(\mathbf{p}) &= T_i(\mathbf{p}, \mathbf{p})
 \end{aligned} \tag{A.2}$$

p_i 에 대한 미분을 편리하게 하기 위해 $\rho(P)$ 에 Cauchy-Schwarz 부등식을 적용하면 S_i 에 대한 Majorizing 함수 T_i 를 얻을 수 있다.

$$\begin{aligned}
 T_i(p_i, q_i) &= \eta_\delta^2 + \eta^2(P) - 2\rho(P, Q) \\
 \rho(P, Q) &= \sum_{\substack{j=1 \\ j \neq i}}^n \frac{w_{ij} \delta_{ij}}{d_{ij}(Q)} (p_i - p_j)^T (q_i - q_j) \\
 &\quad + \sum_{j=n+1}^{n+m} \frac{2w_{ij} \delta_{ij}}{d_{ij}(Q)} (p_i - p_j)^T (q_i - q_j)
 \end{aligned} \tag{A.3}$$

식 (A.3)에서 Q 는 P 와 다른 위치 벡터가 나열된 행렬을 의미하며 q_i 는 Q 행렬에 속하는 I 노드의 위치 벡터를 의미한다. 분산처리 측위를 위해 Q 행렬과 P 행렬은 I 노드의 위치 벡터 이외에 다른 위치 벡터는 같다고 가정하고, q_i 는 이전 iteration의 I 노드의 위치벡터, p_i 는 현재 iteration의 업데이트 된 I 노드의 위치벡터라고 할 수 있다. 주어진 q_i 에 대해 T_i 를 최소화시키는 새로운 위치벡터 p_i 를 구하면 T_i 가 S_i 를 Majorization하여 T_i 의 최소값을 S_i 의 최소값으로 근사할 수 있다. 따라서 아래의 식 (A.4)를 만족시키는 p_i 를 찾으면 된다.

$$\frac{\partial T_i(p_i, q_i)}{\partial p_i} = 0 \quad (A.4)$$

식 (A.4)의 방정식을 정리하면 아래와 같이 나타낼 수 있다.

$$\begin{aligned} & \left(\sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} + \sum_{j=n+1}^{n+m} 2w_{ij} \right) p_i \\ &= \left[\sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} \frac{\delta_{ij}}{d_{ij}(Q)} + \sum_{j=n+1}^{n+m} 2w_{ij} \frac{\delta_{ij}}{d_{ij}(Q)} \right] q_i \\ &+ \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} p_j + \sum_{j=n+1}^{n+m} 2w_{ij} p_j - \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} \frac{\delta_{ij}}{d_{ij}(Q)} q_j \\ &- \sum_{j=n+1}^{n+m} 2w_{ij} \frac{\delta_{ij}}{d_{ij}(Q)} q_j \end{aligned} \quad (A.5)$$

중심 노드를 제외한 다른 노드의 위치는 고정되어 있기 때문에 P, Q 행렬에 속한 모든 노드의 위치벡터에 대해 $p_l = q_l$ ($l = 1, 2, \dots, n, l \neq i$)가 성립한다. 이에 따라 k 번째 iteration에서 $p_l = q_l = p_i^{(k)}$ ($l = 1, 2, \dots, n, l \neq i$) 또한 q_i 는 $(k-1)$ 번째 iteration의 위치 벡터, p_i 는 k 번째 iteration의 위치 벡터이므로 각각 $p_i^{(k-1)}, p_i^{(k)}$ 로 새롭게 표현할 수 있다. 이를 식 (A.5)에 대입하면 p_i 는 다음과 같이 정리된다.

$$p_i^{(k)} = a_i(P^{(k-1)} b_i^{(k-1)}) \quad (A.6)$$

$$a_i^{-1} = \sum_{\substack{j=1 \\ j \neq i}}^n w_{ij} + \sum_{j=n+1}^{n+m} 2w_{ij}$$

$$\begin{aligned} \mathbf{b}_i^{(k-1)} &= [b_1^{(k-1)}, b_2^{(k-1)}, \dots, b_i^{(k-1)}, \dots, b_{n+m}^{(k-1)}]^T \\ \mathbf{b}_i^{(k-1)} &= \begin{cases} b_j = w_{ij}[1 - \delta_{ij}/d_{ij}(P^{(k-1)})] , & j \leq n, j \neq i \\ b_i = \sum_{j=1}^n \frac{w_{ij}\delta_{ij}}{d_{ij}(P^{(k-1)})} + 2 \sum_{j=n+1}^{n+m} \frac{2w_{ij}\delta_{ij}}{d_{ij}(P^{(k-1)})} \\ b_j = 2w_{ij}[1 - \delta_{ij}/d_{ij}(P^{(k-1)})] , & j > n \end{cases} \end{aligned}$$

Abstract

Energy Efficient Localization by Relaxing Node–Connection Density in Wireless Sensor Networks

Kim Changhoe

Department of Electrical and Computer

Engineering

The Graduate School

Seoul National University

Wireless sensor networks with a large number of robots or sensor nodes are studied. The development of low-cost and low-scale technology accelerates the study of wireless sensor networks. Recently, the use of a large number of sensors have gained intensive research. WSNs have many applications in various field. For example, doctor can track a serious patient, dementia patient or other doctors in hospital. In military, bioinspired robots for battlefield surveillance can detect the movement of enemy. In forest, sensor nodes can be aware of fire or protected animal. Localization technique is essential under any circumstance in WSNs. WSNs with a large number of sensor nodes are high density and deployed randomly, so the localization algorithms for these situation must be necessary. And low-cost, low-scale sensors should guarantee low power consumption due to short battery life.

In this paper, we propose two algorithms for energy-efficient localization of sensor nodes by distributed multidimensional scaling in wireless sensor networks. Multidimensional Scaling is the technique to visualize relative distance between the data categorized by several standards. When we define that the standard is distance

between the nodes, we can visualize the location of the nodes by relative distance.

When the node density is high, computational complexity of multidimensional scaling is increasing. As a result, central processing unit (CPU) may be overloaded to carry out the localization in high density sensor networks. And real-time localization become impossible because of delaying the average time for localization. In this paper, two transmit power control algorithms depending on the number of neighborhood nodes which mean the nodes communicating in one hop are proposed. One of those algorithms is that transmit power is reduced discretely depending on the neighborhood nodes. The other is that transmit power is reduced continuously considering the connectivity of edge-positioned node. The root mean square error and power consumption by these algorithms are analyzed in MATLAB simulation result.

Keywords : Localization, Multidimensional Scaling, Multihop Network, Wireless Sensor Networks

Student Number : 2014-21657